An Argus Specialist Publication
April 1985
90p

Your COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE



A MONOPOLY OF GOOD IDEAS FOR THE COMMODORE

Bored with games? Try computer board games

Prism breakout – a modem to make contact with outside

Bank on it – a home accounts program to type in

Our COMMENT

FIRST THE GOOD NEWS. IF YOU OWN A 64, software for your machine is not likely to dry up for some years yet. How do I know? Just look at the recent Commodore launches in the States (see this month's data statements).

The new C128 is compatible with the 64 and can ruin all its software. This alone should be enough to make the C128 a very popular machine – a brand new computer with vast amounts of available software plus an extra 64K. This in turn will extend the life of the 64.

Of course, compatability is a fashionable catchword at the moment. MSX computers have very little else other than compatability to recommend them – and even that's not been 100% proved. They certainly don't offer the latest in up-to-the minute technology. They're chips off a very old block – the Z80.

On the other hand, the C128 is a new and powerful machine (BASIC with over 140 commands, 80 columns, twice the memory) which, nevertheless does not deny its roots. Commodore realises that it is software which holds the key to success for a new machine: a computer which can tap the wealth available for the 64 is already half-way there.

There is a belief, widely held, that as soon as a computer is no longer manufactured, everyone loses interest in it. That's just not true. We know that at Your Commodore from the number of letters we receive from Vic owners asking for more programs, more articles, more anything. But you'll have noticed that software houses are no longer as productive as they once were. You must let them know that you don't want to be forgotten. Only when you stop badgering us will we stop writing about the Vic – and that's a long way off yet.

Plus/4 price slashed

The even better news, if you are planning to buy a Plus/4, is that Commodore have announced a 50% price reduction on this machine. Commenting on this dramatic decision, Howard Stanworth, boss of Commodore UK, said: "At £150, we're offering the public a really powerful and versatile computer at a price which, until now, has only applied to machines designed for first time buyers. We think, at this price, the Plus/4 is head and shoulders above its rivals". He reckons that the Plus/4's in-built software and advanced specification put it in the same bracket as competitive machines costing about £400.

Why this sudden change in pricing policy? "We have always been able to assess and react to rapidly changing situations and the dramatic events of this week are opening up a gap which we are moving fast to fill". What dramatic events could those be? As the saying goes (with, of course, allowance for a little journalistic licence. . .) – 'From little Acorn (financial difficulties) grow mighty. . .price reductions'!

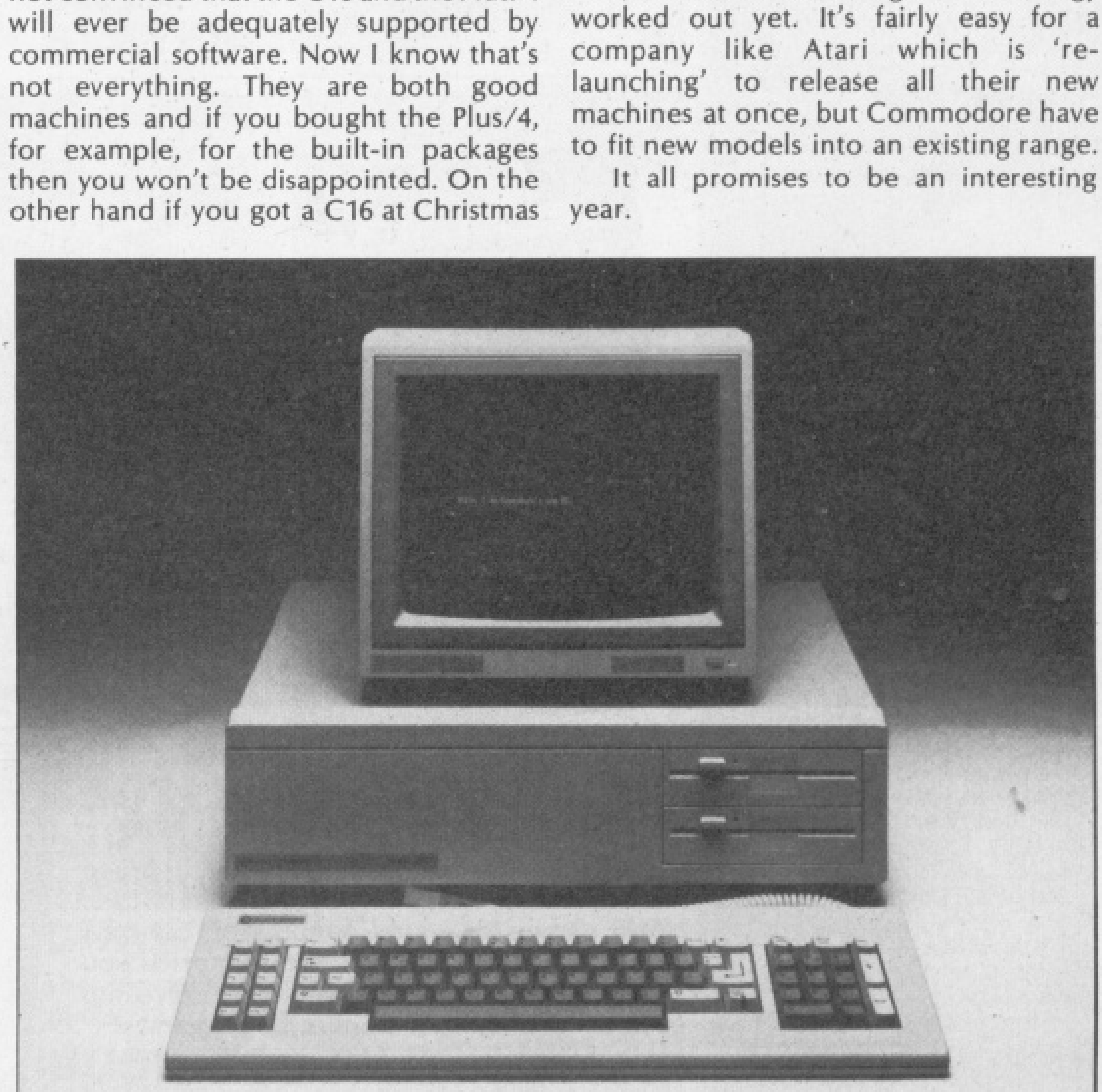
So, what's the bad news? Well, I'm just not convinced that the C16 and the Plus/4 will ever be adequately supported by commercial software. Now I know that's not everything. They are both good machines and if you bought the Plus/4, for example, for the built-in packages then you won't be disappointed. On the other hand if you got a C16 at Christmas because you hoped it would soon rival the Spectrum and the 64 in games, then prepare to find other ways of using your machine to the full. There will be software – after all, software houses can't ignore the 170,000 C16 owners – but most of it will be conversions of the popular games from other machines.

The road ahead

The tide is turning towards more powerful machines and Commodore are inevitably, though not unwillingly, being swept away. The two American machines are examples but there is also the Commodore Amiga which is to be launched this year.

Commodore haven't got their strategy worked out yet. It's fairly easy for a company like Atari which is 're-launching' to release all their new machines at once, but Commodore have to fit new models into an existing range.

It all promises to be an interesting year.





VOLUME 1 NUMBER 7
APRIL 1985
Editor: Wendy J Palmer
Deputy Editor: Kevin Cox
Editorial Assistant: Alison Hjul
Advertisement Manager: Mike Segrue
Advertisement Copy Control: Laura Champion
Publishing Director: Peter Welham
Chairman: Jim Connell
Origination: Ebony Typesetting
Design: MM Design

Editorial & Advertisement Office
No 1 Golden Square,
London W1R 3AB
Telephone: 01-437 0626
Telex: 8811896

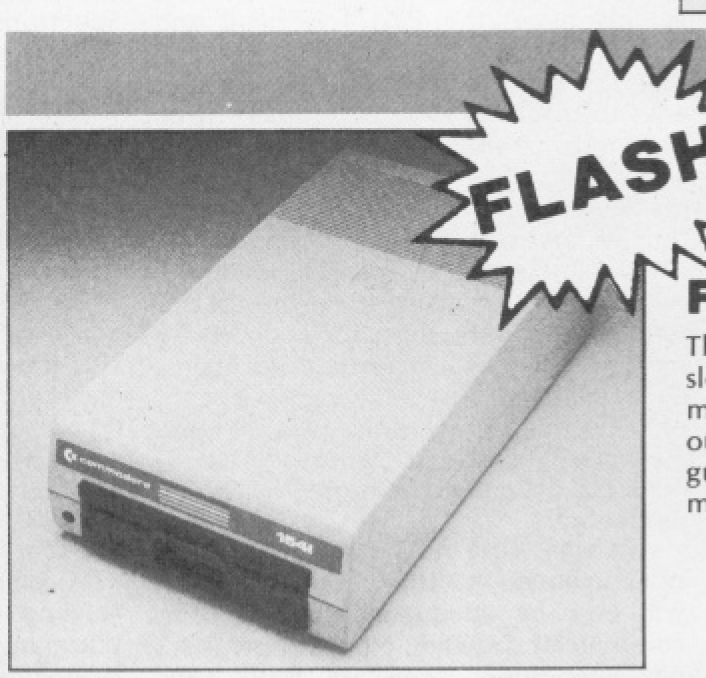
Your Commodore is a monthly magazine appearing on the first Friday of each month.

Distribution by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4JS. Printed by: Alabaster Passmore & Sons Ltd, Tovil, Maidstone, Kent.

Subscription rates upon application to Your Commodore Subscriptions Department, Infonet Ltd, Times House, 179 The Marlowes, Hemel Hempstead, Herts. HP1 1BB.

FEATURES







CONTENTS

SERIES



GAMES AND UTILITIES



REGULARS

GOLD IF YOU YOU H

It's a range of top quality software from Commodore designed to make the most of your Commodore 64's capabilities.

Commodore Gold Medallion Software

New packages will be introduced to the Gold Medallion range every so often, but only if they are really exceptional. They'll definitely be hard soft to beat.

Miss any of them, and you really will be missing out.

MUSIC MAKER

Whether you're an accomplished musician or an out-and-out beginner, Music Maker strikes exactly the right note.

No matter if you've never played a note before, so long as you can hum and you know your ABC, you can start to play famous popular tunes immediately.



And it won't be long before you appreciate Music Maker's many advanced capabilities: you can synthesise many musical instruments, even create your own 'synth' electronic sounds, choose between monophonic or polyphonic play, summon up pre-programmed rhythms and bass accompaniments, and more.

Music Maker is the first in a series of packages which will fully exploit the Commodore 64's outstanding musical capabilities.

On disk or cassette, with music keyboard, a clear and concise manual, and song book, for just £29.95.

SPIRIT OF THE STONES

It's a treasure hunt for 41 real diamonds hidden somewhere on the Isle of Wight.

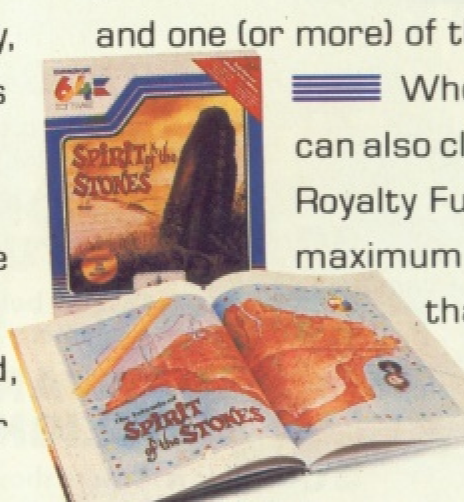
Only one man knows where they are, and he's not saying.

All that he has to say he's said already – but in the form of riddles, clues and puzzles.

Solve the puzzles in the Spirit of the Stones program and you'll find it much easier to solve the puzzles in the Spirit of the Stones book.



Solve the puzzles in the book and one (or more) of the diamonds can be yours.



Whoever discovers a diamond can also claim his or her share of the Royalty Fund, which could grow to a maximum of £1 million. It's a game that's as entertaining as it can be rewarding.

On cassette or disk, £14.99.

INTERNATIONAL FOOTBALL

Already it's recognised as the best football game ever seen outside of Wembley.

It's startlingly life-like, and gives you near perfect control of the players.

You can kick the ball, dribble it, pass it,

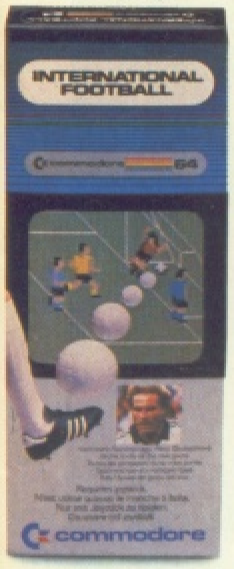
FOR MORE INFORMATION CONTACT: COMMODORE INFORMATION CENTRE, 1 HUNTERS ROAD, WELDON, CORBY, NORTHAMPTON NN7 1QX. TEL: CORBY (0536) 206252

MEDALLION.
HAVEN'T GOT IT
AVEN'T GOT IT.

head it or even throw it in from the touch-line. And every time you belt the ball into the net, the crowd cheer wildly.

How often you score depends on how good you are, and also at what level you choose to play.





There are 9 levels you can play against the computer. Or you can play a friend.

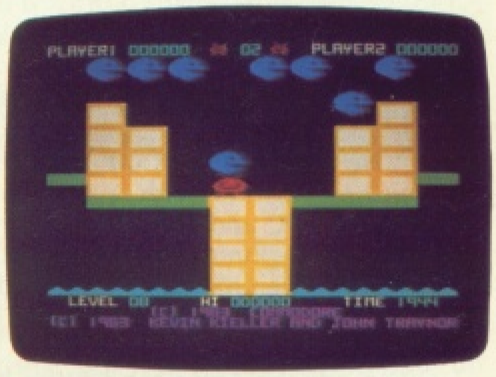
International Football is a real test of skill, dexterity and speed. It is certain to drive you football crazy.

On cartridge, £14.99.

JACK ATTACK

There's no other game like it. It has been voted by the U.S. magazine 'Electronic Games Hotline' as a 'must buy.'

Jack Attack is about squashing heads. It's an addictive game, a game of strategy and cunning. We can almost guarantee it will turn you into a head-case.



To stay alive, Jack must leap aside from collapsing bridges and crashing blocks, and at the same time he must make sure he doesn't fall into the water... and drown.



And that's the easy bit.

Because everywhere Jack goes he's dogged by jolly bouncing sadistic heads that are out to nut him.

He has to squash their heads before his is squashed.

And Jack can't afford to lose his head because we've only given him three, and when they've gone, he's gone...

Jack Attack is available for both the Commodore 16 and Plus/4. On cartridge £14.99.

commodore
HARD SOFT TO BEAT

SELECTED TITLES AVAILABLE FROM BOOTS, WH SMITH, WOOLWORTH, SPECTRUM, JOHN MENZIES, OTHER LEADING RETAILERS AND SPECIALIST COMPUTER STORES.

New PC marks Commodore's return to the business market





COMMODORE'S 16-BIT IBM-COMPATible PC was first shown to a waiting world of dealers, distributors, businessmen and press at the Which Computer Show in January. The new PC isn't actually due for release until late Spring.

No pricing details on the machine have yet been released but Commodore promise that the PC is totally IBM compatible – every test carried out so far has been successful.

David Gerrard, Commodore's UK Marketing manager, believes that Commodore's new assault on the business market is dependent upon building a strong distribution network. Although they don't plan to discard their existing dealers, he says: "We're looking for both new dealers and distributors to expand our range of outlets into a distribution network which is second to none." He hopes to see Commodore hit the business market with a vengeance with the new PC and says "This machine marks the beginning of a carefully-planned programme for Commodore Business Systems, which will take us into the late '80s".

Commodore's American cousins have been far more prolific as shown by the output at the Consumer Electronics Show in Las Vegas in January.

The new American PC is the Commodore 128 which, as the name suggests, contains 128K of memory. With an external RAM Disc Option, the memory can be expanded to 512K. The user has a choice of three operating modes – 128, 64 and CP/M. The 128 mode features BASIC 7.0 which includes over 140 commands, statements and functions. The full 80 column display, along with the extra memory, makes the C128 a credible business machine.

Commodore also announced the LCD portable personal computer. It has 32K of RAM and 96K of ROM and offers eight built-in programs such as word-processing, spreadsheet, address book and calculator. Its major features include a built-in 300 baud modem with accompanying communications software and an 80 column, 16 line, LCD (Liquid Crystal Display) screen.

Also on display was a range of new Commodore peripherals. The 1571 Disc Drive – a 5¼" single floppy disc drive where up to 350K of information can be stored on a single disc – includes a built-in microprocessor, 2K RAM, 32K ROM and transfer rates ranging from 300 cps to 41,360 cps. Other additions were the Commodore 1902 RGBI/Composite Colour Monitor, the MPS 802 Printer and the Commodore Mouse.

But, don't raise your hopes too high. There are no signs, yet, of these CES exhibits crossing the Atlantic to set foot in your local micro stores.

Data Statements

Poking into Prestel



IN THE WAKE OF RECENT, AND MUCH publicised, stories about individuals illegally accessing Prestel databases, we bring you news of yet another 'break-in'. Little did we imagine the outcome when we presented Dave Crisp, a regular Your Commodore reviewer, with a Protek 1200 modem for review in the magazine. He caused concern in the boardrooms at BT and Prestel by 'hacking' into the system.

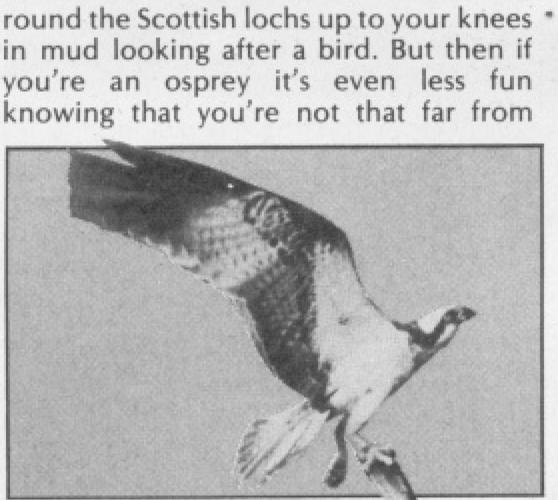
He used his Commodore SX-64 and Protek 1200 modem to dial his local Prestel number and log on as usual. But, this time the response came as some surpprise: he managed to eavesdrop into what he described as 'a response frame for a financial institution' – he had been able to watch people using their building society pages and logging onto the system. For security reasons, he cannot reveal how this was done. But, having successfully repeated the test, Mr. Crisp and his fellow 'hacker', as responsible adults, contacted a security man from Prestel to explain what had happened. Both were invited to London to demonstrate their discovery to the Prestel bosses.

But the weakness lay with British Telecom and not with Prestel. Of course, the implications of what Mr. Crisp did could be very serious. He could access users of any networked system that relies on public telephone lines to transmit data.

Mr. Crisp and his fellow hacker did the right thing in contacting Prestel immediately. By using information gained through such unauthorised access to your personal advantage, you would be liable for prosecution under the Data Protection Act. So be warned.

Give it the bird

IT CAN'T BE MUCH FUN TRAIPSING round the Scottish lochs up to your knees in mud looking after a bird. But then if you're an osprey it's even less fun knowing that you're not that far from



extinction. Just what the osprey and its protectors are up against is portrayed in a new educational package from Bourne Educational Software, called Osprey.

You are given a team of wardens and the responsibility of protecting the small Scottish Osprey population. Out to thwart you are egg-stealers, huntsmen and tourists. The weather's not on your side either which is often the case in Scotland.

Included with the cassette which costs £9.95 is a thirty-two page colour booklet which gives you the background history to the Ospreys in Scotland.

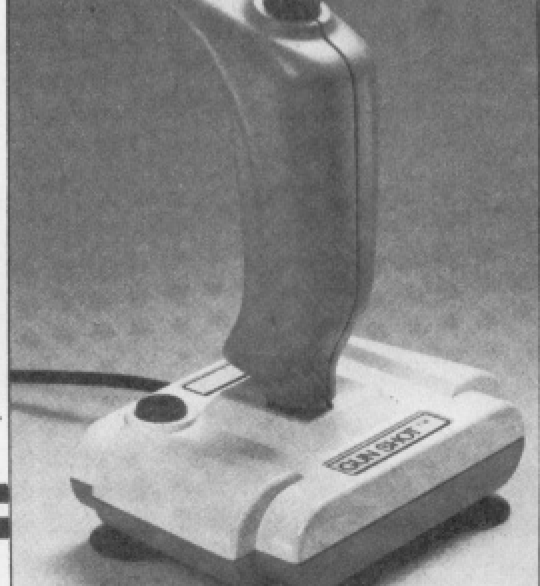
B.E.S. Bourne House, The Hundred, Romsey, Hampshire.

Good shot

Vulcan Electronics claim to have produced a range of 'no-nonsense' add-ons'. At the top of their range is the chunky Gunshot joystick. This joystick is 8-directional and features dual fire buttons, four heavy duty suction pads on the base and a moulded grip handle.

A rapid fire version is also available. The Gunshot costs £8.95 and Vulcan promise an after-sales service and 12-month guarantee. Vulcan are planning to extend their range in 1985 to include the Commodore 16.

Vulcan Electronics, 200 Brent Street, Hendon NW4 ; tel: 01-203-6366.



Ariolasoft in business

ARIOLASOFT UK LTD CONCLUDED AN exclusive agreement, at the CES Show in America, with the American software company, Batteries Included to manufacture and sell its personal productivity in the UK.

Batteries included is a private company and one of North America's top five consumer software manufacturers. The first batch of programs will be launched early summer. They are Paperclip, a word processor package, The Consultant, a data management program, Homepak, a domestic management package and the Home Organiser, a series of dedicated programs designed to solve household and hobby problems. Prices for these titles have yet to be fixed.

Ariolasoft, Suite 105/106, Asphalte House, Palace Street, London SW1E 5HS; tel: 01-828 0720.

Data Statements

Influx from Interceptor

INTERCEPTOR SEEM TO HAVE PREpared a post-Christmas onslaught on the software market, with five new releases for Commodore computers.

First out of the bag is Front Line for the Commodore 64. While in control of the Death Tank Interceptor, your task is to defeat the enemy who control the Edward Isles and recapture the supply dumps.

In Bigtop Barney, as Barney the circus clown you must complete four death defying acts to win the applause of the audience. Barney has four acts to choose from – the high wire act, the monocycle act, the strongman act or the death-defying act of freeing a lion cub from his cage.

The Caverns of Sillahc is an arcade space adventure in which you must guide your ship through the 'Caverns of Sillahc' collecting all your stolen agricultural droids. Your efforts are hindered by numerous hazards and obstacles such as anti-gravity mines, laser bases, antimatter blocks. Ultimately, you must destroy the Sillahc power plant and return to civilisation at the planet surface.



Interceptor's sequel to Heroes of Karn is Empire of Karn. Your task in this graphic adventure is to save the Empire from Zhef and his followers who seek its downfall.

Finally, for the VIC 20, Interceptor have released Villain. As a professional crook, you must avoid the endless pursuits of PC Plodd. It runs on the 8K or 16K expanded VIC 20.

All the Commodore 64 games retail at £7.00 and the VIC 20 game retails at £6.00.

Interceptor Micros, Lindon House, The Green, Tadley, Hants. Tel. 07356-71145.

From cult to occult

• CRL HAS JOINED FORCES WITH Richard O'Brien of Rocky Horror Show fame to produce a computer version of the cult show. There are very few details yet on content or price but the game should include fully animated character from the show and Rocky Horror mus. It should be available in the late spring.

CRL, CRL House, 9 Kings Yard, Carpenter's Road, London E15 2HD; tel: 01-533 2918.

• BUBBLE BUS HOPE TO MAKE A SPLASH with their latest release for the Commodore 64, Aqua Racer. In your Aqua Racer, your task is to beat the opposition while sticking close to the bends and narrowly missing the jagged rocks.

The game is written in machine code and includes 3D graphics, specially composed music, 20 different courses and joystick control. It retails for £6.99.

Bubble Bus Software, 87 High Street, Tonbridge, Kent TN9 1RX; tel: 0732-355962.



• WITCHSWITCH IS THE LATEST RELEASE from the English Software company. It has been designed by the Psychic Engineering Control Group and tells the story of the race against time to save a village from destruction by molten lava. You must reverse the flow of the lava before it's too late! Witchswitch sells for £8.95.

The English Software Company, Box 43, Manchester M60 3AD; tel: 061-835 1358.



Data Statements

School winners



Commodore hope to be top of the class with their "Computers for Schools" sponsorship scheme to enable schools to purchase selected computers and peripherals at 30% discount.

Although the sponsorship scheme was initially designed to help schools entering the 1985 Commodore/BCS Computer Quiz to buy their own computers, the scheme is now being offered to all UK schools to help them beat the education cuts. Every £7.00 a school raises through sponsorship, Commodore will match with £3.00. For example, to purchase a Commodore 64, a school would have to raise £139.00 excluding VAT and Commodore would pay the difference on the recommended retail price of £229.00. The scheme is valid for the Commodore 64 and its peripherals and the closing date for receipt of completed sponsorship forms is 30 April 1985.

For further information, contact Aileen Bradley at Commodore Business Machines (UK) Ltd, 1 Hunters Road, Weldon, Corby, Northants, NN17 1QX; tel: 0536 205555.

Commodore are also offering a special deal on the 8296D business micro. This normally retails for £1,690 excluding VAT but Commodore are offering it to accredited educational establishments throughout Britain for £995. Each machine also includes three software packages – Superscript, a word processing package, the Manager (database and file management package) and Calc Result (financial planning spreadsheet). Those interested in this special offer should contact the Commodore Information Centre on 0536 205252.

A special area of Compunet has been set aside for educational software – schools with one of Commodore's modems can choose from over 60 free educational programs. Commodore 64 Modem users can also access Prestel and a wide range of other viewdata services, as well as direct 64 to 64 communications.

More Melbourne Magic



THERE ALWAYS SEEM TO BE SINISTER and mysterious goings-on at Melbourne House. Following in the footsteps of the haunting Castle of Terror and magical Zim Sala Bim, come their latest offerings – Dark Tower, Grand Larceny and Disc Hobbit. These are the first of their new releases for the Spring which also include conversions of Wizard and the Princess and Classic Adventure for the C16.

In the Dark Tower, Prince Harry (not him of the royal blood and jug-ears) has been turned into a mutant by the guardian of the Dark Tower. As Prince Harry, you are trapped in a series of chambers within the tower and, to turn back into a prince, you must overcome your evil keeper's automatic defence systems, collecting all the jewels in the tower and delivering them to the guardian. Having done this, you will be allowed into the final chamber where the secret of the tower is revealed and you acquire great wealth. There is also a puzzle to solve, to which clues are offered throughout the game. If you solve the puzzle, you might win a Melbourne House game of your choice.

Grand Larceny is an animated graphic adventure. You must recover the secret plans (hidden inside a hotel) which are to be smuggled out of the country without the enemy agents catching you.

The Hobbit is a Melbourne House classic and is now available on disc. As Bilbo the Hobbit, you must steal the dragon's treasure while encountering many strange characters and dangers on the way. The disc version has more detailed graphics than the original and, so its makers claim, includes illustrations the like of which have never been seen before on a home computer.

Grand Larceny and Dark Tower cost £7.95 and The Hobbit cost £17.95.

Melbourne House, Castle Yard House, Castle Yard, Richmond, TW10 6TF; tel: 01-940 6064.

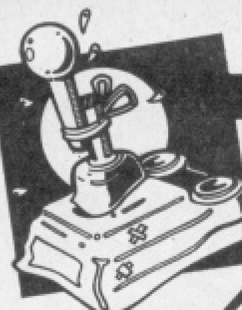
Note on Program Listings

DUE TO COMPLAINTS ABOUT illegible graphic characters in the program listings, programs in Your Commodore are now listed with a REM statement inserted before each line containing graphic characters. We have used a Your Commodore program (All Clear, January issue) to do this. Because we do not alter the program line numbers when we insert the REM statements, many of the line numbers are duplicated. However, the REM statements are merely there as a guide: they do not affect the running of the program and, therefore, there is no need to type them in.

Errata

WHOOPS – WE'VE BOOBED AGAIN! TEN out of ten for all those readers who noticed the three missing lines in Laser Track (January 1985, page 30). In part 3 of the program listing, lines 8400, 8305 and 8310 should be as follows:

```
8300 POKES1,128 : POKES2,12 :
POKES3,RR : POKES1,129
8305 POKES1-7,128 : POKES2-7,44 :
POKES3-7,5 : POKES1-7 129 : POKEV+30,0
8310 RETURN
```



IN ARCADIA



Having already catered for adventure buffs, we now hope to satisfy arcade fans with this new series from the man with his finger on the button, Phil South.

I TRUDGED PAST THE ARCADE THIS afternoon because I was skint. They've just got that new game in, too! It's called I, Robot, and it uses ultra-hi-res 3D supercomputer graphics, none of that stupid laser disc caper ... I felt really depressed, so I dashed home to work off a bit of my frustration on some of my favourite games. Imagine my surprise when I was greeted by a parcel of new games for my scrutiny; yeehoo!!! I can't help it, you know, I'm an incurable fanatic. (It all started with electronic ping-pong, and it's been downhill ever since.)

Having sat down at the keyboard and powered up my '64, I loaded up the first out of the bag. Space Shuttle (Activision) is a sort of flight simulator for space-race junkies. You take off from Cape Canaveral on a sunny afternoon, and blast off (with suitable rumblings and vibrations) into outer space; your mission (should you decide to accept it) is to 'rendezvous' with a renegade satellite and bring it back to the good old USA. I thought this game was great, although a trifle easy once you get the hang of it; I played it repeatedly until I got bored. The shuttle movements are quite limited and, unlike most ordinary flight simulators, you can't crash the shuttle back through the atmosphere until you've got the satellite. The graphics are a bit chunky, and the sound is never more than just appropriate, but it's great fun.

Star item

By far the best of the crop this month is, predictably, that mega-media experience, Ghostbusters (Activision). It's a fact of life these days, if you make a pop song, you gotta make a video; if you make a movie, you gotta make a videogame! I can't leave this thing alone, and it's driving me nuts. I wake up in the middle of the night saying "aha, if I buy the compact car and capture as many slimers as I can then I'll get to the end screen". It's crazy, but I love it. The plot follows the movie exactly, right down to screaming "he slimed me" and shouting "Ghostbusters!" at the appropriate moments in the soundtrack. The music, sound effects, graphics and general 'fun-to-be-had' puts this game head and shoulders above the rest. Put-downs of commericalism aside, this is a first rate game and as such gets my 'hot game of the month' award. Get it or regret it!

If you do well in Ghostbusters, you are assigned a bank account number, and this allows you to begin at some other time with a larger amount of money. As my special gift to you, no charge, I can supply you with my own account number for $16,000!!! First, when it asks you for your name, type SOUTH then (RETURN), then my account number 40012500. The account number is generated from the name and the figure, so you must type the same name as I did. (For the hedonists among you, there is an account number which allows you over $67,000. For name type in (RETURN), and the number is 66455701 (Special thanks to my mate Neil Davey for that one.)

Other gems

The itinerant Cuthbert in the Tombs of Doom (Microdeal) is next up, and very jolly it is too. I never took to old Cuthbert, I rather thought him a bit of a wally, but I must say I do like his new game. Perhaps a touch too much like Quicksilva's Fred; you know, tramping around 200 or so tombs in the footsteps of Indiana Jones, picking up treasure and getting murdered by spooks, bats, devils, sworded spheres, vicious psycho-saxaphones, etc., but a nice little arcade adventure when all said and done. Nothing stunningly different about the look and sound of this one, but it does keep me playing, which must, I suppose, say something for its addictive qualities.

Summer Games (Epyx/Quicksilva) is fantastic. It's one of those Olympic or decathlon-type thingies, and as such should be unremarkable. BUT, the graphics are superb, the sound is pure realism and the games are fun and challenging. It's a hit, and as long as my joystick holds out I will be skeet shooting, pole vaulting, swimming, diving, running, and doing Nadia Komanechi impressions!

In the bin

There had to be some duff ones, and here they are. Roulette (No Man's Land) is a real throwback to the all-text games of the infancy of computer gaming. I've got a very low boredom threshold, and this turkey definitely overstepped it! Magic Micro Mission (Quicksilva) is supposed to be for kids, but most of the kids I know wouldn't give this one a second glance, TV program tie-in or not. Seeing Orpheus in the Underworld (Sterling Software) just goes to prove I know a rip-off of Pitfall when I see one, and all games with music in them should have the facility to toggle it on and off!

Letters department

I have a note here from Chris Charry of Co Antrim, asking me to list the best versions of virtually every type of game in existence! What I do have space to say is Flight Simulator II (Sierra), Android 2 (Vortex), Boulderdash (First Star), Catacombs (Anirog), Midway (PSS), and literally any text graphic adventure by Melbourne House. I phoned Ultimate on your behalf, and they say that they "have no plans for converting any games at the moment", which is a bit strange, as I didn't even tell them which ones I was talking about. Still, that's their problem. As for the Ant Attack query, how the dickens did you manage to burn your instructions before you played the game? Nice graphics, shame about the game.

That's all South, get off!

OK, that's enough blabber from me. Turn to this column again next month for more up-to-date comment and answers to your nagging queries; Flippo has the technology!

Instead of ten aliens, Cl
for his Commodo

What happened ne



Saturday morning.

Waltzed into my local computer shop.

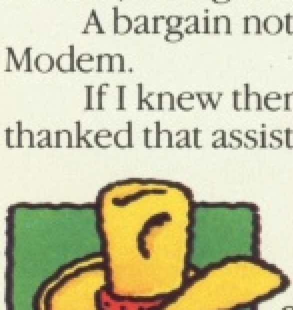
Packed as usual with masses of kids enjoying the arcade games.

Surely I could put my Commodore 64 to better use. Helpful assistant suggests a Commodore Communications Modem.

Tells me it comes with a year's free subscription to Compunet, a new network service, saving me a cool thirty quid.

A bargain not to be missed, so I bought a Modem.

If I knew then, what I know now, I'd have thanked that assistant more.



Saturday (one week later).

Fantastic.

My Compunet membership came through this morning.

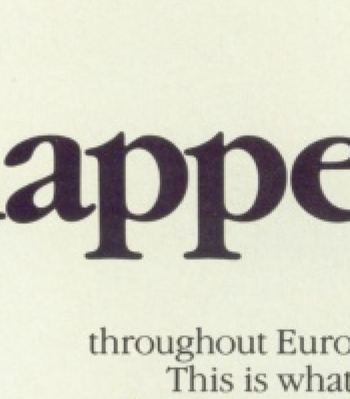
Hurriedly plugged the Modem into my 64's cartridge port, and hooked up to the telephone line.

Can't wait.

At last I can communicate with other Commodore 64 Modem owners and giant mainframes.

What's more, I can also access databases throughout Europe and the U.S.A.*

This is what home computing's all about.



Sunday morning.

Raining.

Tapped in my Compunet I.D. and personal password.

Wow, what a directory!

Decide to pit my wits against other Modem users by entering Multi User Dungeon, an interactive on-line game.

Should stretch the old grey matter a bit.

Then a quick look in 'The Jungle.' This is an open area where other Modem owners display messages.

See a Commodore user in Fife wants to sell 'U-boat' for £3.00.

Leave message offering him 'Mighty Gork' on a straight swap.



Monday evening.

Move on to the Compunet Software Park.

What a choice. Loads of high quality bargain programs.

Particularly interested in educational software, so I call up 'The Study.'

Download free physics package to help with my exams.

Clive bought a Modem
lore 64.



ext changed his life.



<u>Tuesday evening.</u>

Dad's turn.
I don't get a look in as he's busy teleshopping.

Actually it's amazing what bargains turn up. He even finds a new house.

Mum said she doesn't want to move and anyway his dinner's getting cold.



<u>Wednesday evening.</u>

Discover I can join BLAISE,* the computer service for the British Library.

Their catalogue of books dates way back to 1950.

Should give me an interesting edge over my school chums.



<u>Thursday evening.</u>

Sis has a go.
She keys into Prestel.*
Imagine, over 300,000 pages of information and news.

What does she choose? The lonely hearts section.

She's disappointed. Couldn't find Simon le Bon's private number.



<u>Friday evening.</u>

Yippee! Receive a reply from the guy in Fife.

He fancies taking on Gork.

What's more he's written a program he'd like my opinion on.

He transfers it direct, using the free user to user software.

I've made my first computer pal.

It has really been a week.

Best one I've had since getting my Commodore 64.

Sure am glad I got the Modem instead of all those aliens.

The Commodore 64 Communications Modem comes as a complete package with a year's free subscription to Compunet, for just £99.99 inc. VAT.

Find out how a Modem can change your life. See it now at Dixons, Curry's, Comet and selected Commodore dealers.

commodore

For further information phone or write to:
Commodore Communications Modem,
1 Hunters Road, Weldon, Corby,
Northamptonshire NN17 1QX, Tel: 0536 205252.

*Require additional subscriber charges. Prestel is a registered trademark of British Telecom.



LIGHTNING STRIKES AGAIN

BASIC LIGHTNING

WHITE LIGHTNING

MACHINE LIGHTNING

OASIS SOFTWARE

THREE HIGH POWERED GRAPHICS DEVELOPMENT SYSTEMS FOR THE **COMMODORE 64.**

Totally dedicated to writing very fast, video games, BASIC Lightning is a is a fully structured extension to the Commodore BASIC which adds a staggering 200 reserved words. It allows up to five tasks to run concurrently (one in foreground and four in background). Most of the commands are dedicated to games writing and the sound and graphics commands are unparalleled. Procedures and PASCAL type structured programming commands are also a feature of BASIC Lightning.

As well as the Commodore's own 8 hardware sprites, BASIC Lightning has its own software sprites. Up to 255 can be defined with user selectable dimensions. These can even be several screens wide. They can be scrolled, spun, mirrored, enlarged or inverted with phenomenal speed and smoothness.

A Sprite Generator Program (written in BASIC Lightning) is also supplied and can be used to design, edit and store all your sprites for use in your main program.

White Lightning is a complete games writing package comprising a high level, Forth based, multi-tasking games writing language, the extended BASIC (see BASIC LIGHTNING) and a powerful sprite Generator Program. Programs can even be written in a combination of Forth and Commodore BASIC and the final program, which will run independently of White Lightning, can be marketed with no restrictions whatsoever.

The Basic Lightning part of the package can be used to experiment quickly and easily before the Forth program is developed.

The speed of White Lightning has to be seen to be believed and a full demo is included. As with Basic Lightning, hardware sprites are supported, together with 255 software sprites which can be scrolled, spun, reflected, enlarged or inverted.

MULTI-TASKING Without doubt the most powerful feature of the Lightning series of languages is the multi-tasking facility. This allows two programs to be run concurrently and makes those smooth landscape scrolls etc. effortless.

The BASIC Lightning Sprite Generator Program is also included in the package.

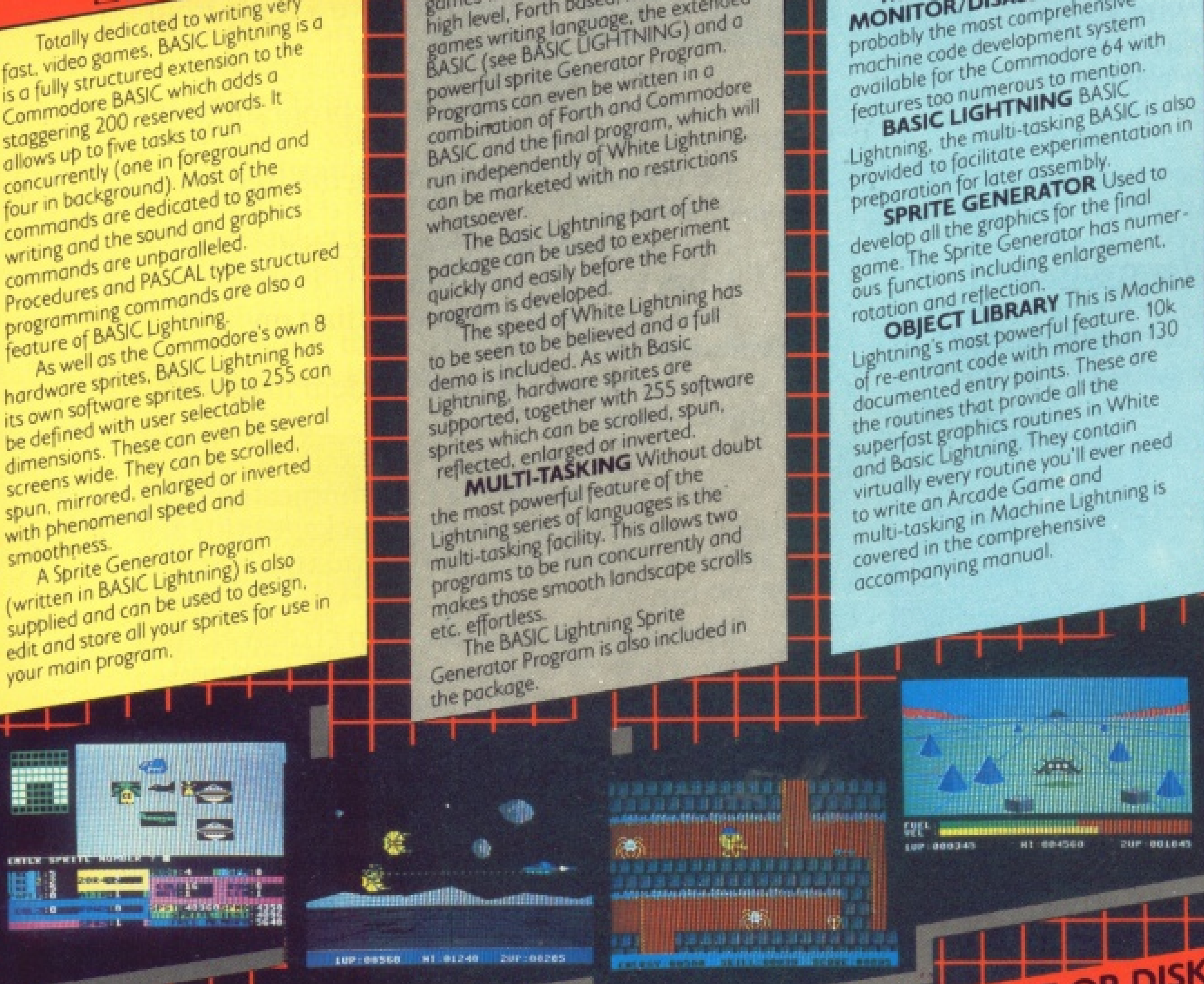
Commodore 64 Machine Lightning is probably the most advanced games writing utility available on any micro. It comes in 4 parts:

MACRO ASSEMBLER/ MONITOR/DISASSEMBLER This is probably the most comprehensive machine code development system available for the Commodore 64 with features too numerous to mention.

BASIC LIGHTNING BASIC Lightning, the multi-tasking BASIC is also provided to facilitate experimentation in preparation for later assembly.

SPRITE GENERATOR Used to develop all the graphics for the final game. The Sprite Generator has numerous functions including enlargement, rotation and reflection.

OBJECT LIBRARY This is Machine Lightning's most powerful feature. 10k of re-entrant code with more than 130 documented entry points. These are the routines that provide all the superfast graphics routines in White and Basic Lightning. They contain virtually every routine you'll ever need to write an Arcade Game and multi-tasking in Machine Lightning is covered in the comprehensive accompanying manual.



AVAILABLE ON TAPE OR DISK

Available from Boots, John Menzies and all good software suppliers.

OASIS SOFTWARE

9a Alexandra Parade, Weston-super-Mare, Avon BS23 1QT Telephone: (0934) 419921.

Every product carries a lifetime guarantee.
All prices include comprehensive manuals, VAT and p&p.
(includes manuals and p&p overseas)

24-hour Access Tele-ordering on (0934 419921)

Please send me the following Lightning System Pack(s)

Basic Lightning (tape) £14.95 ☐
(disk) £19.95 ☐
White Lightning (tape) £19.95 ☐
(disk) £29.95 ☐
Machine Lightning (tape) £29.95 ☐
(disk) £39.95) ☐

I enclose my cheque/P.O. for £________

Name________
Address________
____
____
Telephone________

You don't have to link up to Compunet to discover that Modem magic. Dave Crisp and Henry Budgett discover worthy alternatives in the Protek and Prism moems.

IT'S FOR YOU-HOO

The first thing I did was to get the tape-based software onto disc because, as all the regular readers know, I have an SX64 and so can't use a tape. Loading the tape is slow so, if you do have a drive, it would be worth doing anyway.

The interface pack (available separately) contains the software and also a lead to connect the user port and the RS232 socket on the modem.

If you are logging onto Prestel, you enter your 10 digit ID which is stored. You are then prompted to dial Prestel. It is worth noting that, if you have had your Prestel number for a while and your call was not a local one, things are rapidly changing and Prestel now claims that 92% of people can connect with a local call so check up as you may be spending more than you need.

An earpiece plugs into the modem so you can hear when the Prestel computer answers. A touch of the F3 key and your 10 digit ID is sent to the computer. Procedure from then on is like standard Prestel.



In and out

It is possible to return to the main menu while still logged on. This is achieved by pressing the F1 key. From the main Protek menu you are able to:

1. *Save Frame*

You can return to the menu and save the current frame on tape or disc. This is particularly useful for things such as messages and timetables. It is not possible to overwrite an existing frame with one of the same name, thus preventing mistakes.

2. *View Saved Frame*

You can view any saved frame without logging off, as long as you know the frame's name. If you don't, there is no way of getting a directory of the disc without newing the program. This is inconvenient to say the least. Loading a saved frame is fast compared to many communications packages so, if you are looking at a frame while logged on, you are not burning pounds.

3. *View Current Frame*

The last frame displayed while logged on is held in RAM even after you have disconnected and so, if you are looking at a specific page, you can disconnect and study it at leisure with no connection cost. Of course, if you log off properly via page 90 of Prestel that is the retained frame and the log-off frame is about as exciting as watching the bath empty!

4. *Change ID*

You can change ID without unplugging and starting again. This is useful as you only know if you have input the correct ID when you try to log on. This gives you the opportunity to change it.

5. *User to user*

This allows you to connect up with another Protek user in order to transmit

software to each other but I will deal with that separately.

Corruption

Because this is an acoustic modem you are prone to suffer from the occasional corrupt screen. This appears as squiggles and dots in place of letters. The best way of doing this is to get people to call your name or smash cups of coffee onto the table. To avoid it, I have made a very simple acoustic box. This consists of a shoe box covered in foam which I put over the modem after placing the modem on another piece of foam. Crude but very effective.

You can also get garbage on the screen from the various buzzes, clicks and bangs that Telecom seem to introduce themselves. If you find you are getting a lot of corruption listen through the earpiece and, if you seem to have a particularly noisy line, it may be worth disconnecting and trying again.

I have tried a couple of other acoustic modems in the past and this one does seem less prone to external noise. The rubber cups that the phone fit into are nice and tight and keep out all but the loudest bang.

So far so good

The modem requires four pencil batteries which seem to last well despite being left on overnight on many occasions. It is also portable. Providing that the place you use it has a standard telephone handset (not the trimphone type) you can plug in anywhere (even a phone box).

It must be said that the modem itself is uncluttered by switches. There is the RS232 socket, an earpiece socket and one switch all at one end. The switch is a three position switch which is either (a) off, on, and in 1200/1200 mode or (b) on, and in 1200/75 mode. A small red LED indicates whether or not the batteries are up to scratch. The 1200/75 switch is used for Prestel and a few other boards and the 1200/1200 is mainly used for user to user.

Some of you may have noticed that there is no 300 option. 300 baud is favoured by many of the advertised bulletin boards. A few do work on 1200/75 but they are in the minority. Due to this I feel that this is best considered as a cheap Prestel link.

Bugs

I have grown fond of this modem. Most of the time it works very well and without too many problems. It does, however, have the annoying habit of suddenly cutting you off. Why it does I do not know and will endeavour to find out. Usually it happens when I have typed out a message for a person and try to refresh the screen display with ★00#. There is a sudden click, the screen clears, and the line goes dead. There is no alternative but to re-run and phone again. Annoying.

Talking of leaving messages, the most irritating bug is the one that leaves you wondering what you have said. Let me expand. If you are leaving a Mailbox for another Prestel user you type in what you want to say and the on-screen display should show what you have typed. Not so. The cursor leaves behind it a trail of white squares. This means that you often end up with spelling mistakes and word ends, split between two lines. I feel that whenever I leave a message for someone I must tell them about the bug or they think I am illiterate!

Protek user to user

The software for user to user communication is separate from the main software and loaded from the main program. There are two modes – transmit and receive. You are then prompted to say whether you will transmit (receive) BASIC or code. After putting in the start address and the number of bytes to be transmitted you are prompted to transmit. If the header becomes corrupt or the file becomes corrupt you are informed at the end and prompted to transmit again. It is not a bad piece of kit and works well. The unit is let down by its software but that should be remedied in the future.



Apparently this is due to Prestel echoing the character you have typed and confuses the 64 and results in a white square or some such blurb. This information came from Protek and they say they are working on it.

Apart from that there is little trouble with the modem. Its facilities are basic to say the least but I am sure that, because most of the problems are software based, they could be rectified. If you come up with any modifications or improvements to the software why not let us know at Your Commodore.

Maybe it would be possible to prepare messages while logged off and then put it onto Prestel as a complete page already typed and checked. That way it would not matter if it did look like nothing on earth as you would already have been able to edit it.

Checking with Protek they are at least aware of the problems and acknowledge the bugs. If these things are sorted out I should think they will have a very popular product.

Technical specification

The input socket in the modem is standard RS232. The pins are as follows:
PIN 1. GROUND
PIN 2. SERIAL IN
PIN 3. SERIAL OUT
PIN 4. 5V
PIN 5. ANSWER/ORIGINATE (non standard use).

By the way if you are a Commodore user why not mailbox me. My systel number is 106434851.

Protek 1200 modem
£59 (plus £10 for an interface)

Protek Computing Ltd.
1a Young Square
Brucefield Industrial Park
Livingstone SH54 9BX.

Prism

Although Commodore and the Press are making a lot of fuss, and in many cases rightfully so, about the Compunet data base system many users might want to get at other systems like Prestel, Micronet or even a little user to user communication. Prism, who have long been involved in bringing modems to the masses, have put together a rather neat little package based on their direct connect Modem 1000 together with a plug-in cartridge from OEL which effectively turns the Commodore 64 into a Viewdata terminal.

Because the modem is a separate unit designed to work with other computers as well as the Commodore 64 the overall appearance of the package isn't terribly neat although the component parts themselves are by no means ugly. The modem is a small (165mm by 50mm by 235mm) black box with three indicator LEDs and two switches at the front and the power and 'phone leads to the rear with a couple of sockets for the serial interface and telephone handset. The front panel indicators provide an indication of power to the unit, carrier detected and whether the unit is on-line. The on-line switch is fairly confusing – should it go down to connect or up . . . experimentation reveals the answer but better instructions in the manual would help. The only other control selects the mode of operation; Viewdata (1200/75) or 1200bps in either Receive or Transmit.

Internally the modem is rather a disappointment. Constructed on an SRBP PCB board rather than a fibreglass one, cost-cutting in the extreme, the circuitry is not based on the World Chip modem system but on the Viewdata modem chip which explains why there's no 300bps option. The layout is also very generous; the unit could have been some 30% smaller with ease. The construction is solid enough with only the PCB-mounted sockets likely to cause trouble in the long term.

Communications pack

The communications pack, a neat 150mm by 45mm by 115mm, plugs into the Commodore's cartridge slot. Interestingly the internal construction of this unit is much higher than that of the modem yet they are both made by the same company. The modem connects to the communications pack through a special cable from the serial port. Your telephone must be unplugged from the wall. If you don't have the new style plugs you'll have to get one fitted, and the modem cable plugged in its place while the phone now connects to the back of the modem. This is the only modem I've used which won't allow the use of a 'two-into-one' connector, although there is a certain logic in their method since it means you don't need to have an extension socket.

You must now try to disentangle the wires to the rest of the system and actually get to the keyboard in order to get connected. The OEL Communications Pack makes this really easy although the manuals that come with this and the modem are really pretty awful. A first time user could get pretty confused by some of the contents and a re-write wouldn't go amiss. Nevertheless, once the system is connected and turned on the whole thing is menu-driven and very easy to operate.

Running the system

Getting the system up and running is pretty easy, given that you have a Prestel account number and password handy. Unlike Commodore's system there is no 'fingerprint' built into the system and so any valid password can be used. Hackers please note that Prestel's security has been tightened considerably in recent months! (That's what he thinks; see Data statements – ed.) The legitimate user will find little trouble in registering the system for use as Prism seem to provide all the necessary forms as a matter of course although in our case they let us borrow their account number – trusting, aren't they?

Apart from being able to access any of the usual Micronet and Prestel pages the OEL software can cater for downloading software and storing it onto tape or disc. Ordinary pages can also be saved in this way but with the speed problems of both Commodore's tape and disc systems there cannot be much advantage unless you want to view something again and again. If you've got a printer attached the software will also allow you to dump pages onto paper. Sadly, I wasn't able to test this feature because my printer isn't Commodore graphics compatible. But, given that it supports Prestel-type block graphics it could be very useful for dumping messages and the like. With messages, the software allows you to prepare and edit plain text in an off-line mode which helps save on connection time.

Prism user to user

One feature not supported by the Compunet system – at least not in the usual sense – is user-to-user communication. Compunet users can place information in the 'Jungle' or send messages in much the same way as the Prestel user but owners of the Prism/OEL system can indulge in real-time communications. Admittedly the connection can only be made in one direction at a time so you do have to change the modem from 'send' to 'receive' but that seems a small price to pay for the speed and convenience of the facility.

As far as I could tell from the documentation there is an error checking facility provided which works on a block by block basis to check that what was sent really got there. Christensen protocol is so widely supported by communications packages and bulletin boards that it would have provided a bonus had this been offered as well. But, at least OEL provided some form of checking. Whether the protocol they use is in any way Christensen compatible I cannot say.

In conclusion

The Prism Modem 1000, in itself, isn't terribly special. Had it offered a 300bps full duplex speed as well it could have been used for bulletin board access and real user to user communication. Given that it doesn't possess this option, it performed very well indeed. Possibly the only item that could do with being tidied up is the labelling of the on-line switch as neither the operating manual or the front panel labelling make clear which way is 'on-line'.

The software package, on the other hand, is very user friendly and requires little real effort to understand. Sadly this too is let down by the documentation but the screen directions are usually clear enough to get around most of the potential problems.

Overall, if you are looking for a communications system that offers Prestel/Micronet access and the chance of doing some program sharing down the line the Prism package isn't at all bad. But, for those determined to have a crack at Compunet, or who want to hack away at the real bulletin boards, it is not the best option by any means.

Prism Modem 1000
£69.96 (£129.95 for complete package)

Prism Micro Products Ltd.
Prism House
18/29 Mora Street
City Road
London ECIV 8B7

If you are thinking of joining Prestel you can either call them on FREEPHONE 2355 or write to:

Prestel.
Telephone House
Temple Avenue
London
EC4Y 0HL

RESTORE TO LINE

Mike Hart shows 64 users how to restore the pointer which reads data statements to a particular line or data statement.

```
1000 REM * RESTORE TO LINE:3 VERSIONS *
1010 :
1020 REM    ***  BY M.C.HART  ***
1030 :
1040 REM TIMINGS HAVE BEEN MADE OVER
1050 REM 2000 DATA STATEMENTS ON A C-64
1060 :
1070 :
2000 REM  RESTORE (SIMPLE)
2010 :
2020 DATA ***,===
2030 :
2040 TI$="000000":FORJ=1TO1E6:READ X$:IF X$<>"***"THEN NEXT
2050 J=1E6:NEXT:T=TI/60:READ X$:PRINT X$,T"SECS"
2060 :
2070 :
3000 REM  RESTORE (BASIC)
3010 :
3020 LP=2049:TL=2020:TI$="000000":REM TL=TARGET LINE - LP ALWAYS STARTS AT 2049
3030 FORJ=1 TO 1E6
3040 IFPEEK(LP+2)+256*PEEK(LP+3)<TL THEN LP=PEEK(LP)+256*PEEK(LP+1):NEXT
3050 J=1E6:NEXT:LP=LP-1:POKE66,LP/256:POKE65,LP-PEEK(66)*256:T=TI/60
3060 READ X$:PRINT X$,T"SECS"
3070 :
3080 :
4000 REM  RESTORE (MACHINE CODE-C64)
4010 :
4020 REM  SYNTAX = SYS LOC'N(TARGET) OR SYS (LOC'N)TARGET  ... NO COMMA!
4030 :
4040 READ X$
4050 FORJ=828 TO 851:READ X:POKEJ,X: NEXT:DATA 32,138,173,32,247,183,132,95,133
4060 DATA 96,32,19,166,164,95,208,1,202,136,132,65,134,66,96
4070 :
4080 TI$="000000":SYS 828(2020):T=TI/60:READ X$:PRINT X$,T"SECS"
4090 :
4100 REM VIC20 : CHANGE BYTES 3,6,13 TO
4110 REM         205,215,198
4120 :
4200 REM BASIC 2 :DATA 32,139,204,32,210,214,132,92,133,93,32,44,197,164,92
4210 REM BASIC 2 :DATA 208,1,202,136,132,62,134,63,96
4220 :
4230 :
4400 REM BASIC 4 :DATA 32,132,189,32,45,201,132,92,133,93,32,163,181,164,92
4410 REM BASIC 4 :DATA 208,1;202,136,132,62,134,63,96
4430 :
4440 :
5000 REM *** TIMINGS - C64 (N=2000) ***
5010 :
5020 REM SIMPLE ....       9.6 SECS
5030 :
5040 REM BASIC  ....       5.4 SECS
5050 :
5060 REM MACHINE-CODE ..   0.02 SECS
READY.
```

I have written three different types of Selective Restore each of which restores the pointer that reads data statements not to the first one but to a particular line or a particular piece of data. Each is presented below, together with references to the program lines in the composite program and timings are given for the technique when 2000 data statements have to be scanned.

Method 1: Simple (Lines 2000-2070)

In this method you use a very large loop (i.e. J set to 1,000,000) to locate a data marker which immediately precedes the item of data that you wish to access. In this case, the data marker is the string "★★★" and the first piece of data is the string that follows, i.e. "===". This technique is the longest (nearly 10 seconds) but the easiest and will suffice when the data items is not very many. It is very easy to program.

Method 2: BASIC (Lines 3000-3080)

This method requires you to specify your target line in a variable (which here I have called TL, i.e. line 2020). LP is used as a line pointer to step through the program and should be initialised for Commodore 64/VICS to 2049 and for PETS to 1025. Again a very long loop is chosen and the program 'steps through' the line numbers found in the third and fourth bytes of each line as it is internally stored in order to find the line required from which to read data. When found, appropriate values are POKEd into the zero page pointers which keep track of data statements. The timing is about half that of Method 1 (about 5 secs).
NB: For PETS, substitute 63 for 66 and 62 for 65 in line 3050

Method 3: Machine Code (Lines 4000-4440)

This 'machine code' version is stripped down to make it as compact as possible. It is located in the cassette buffer although it can go into any area of protected memory. To make the routine as compact as possible no provision has been made for the UNDEFINED STATEMENT ERROR if the sought line does not exist – instead one may get an OUT OF DATE ERROR. Notice that the syntax is *either:*

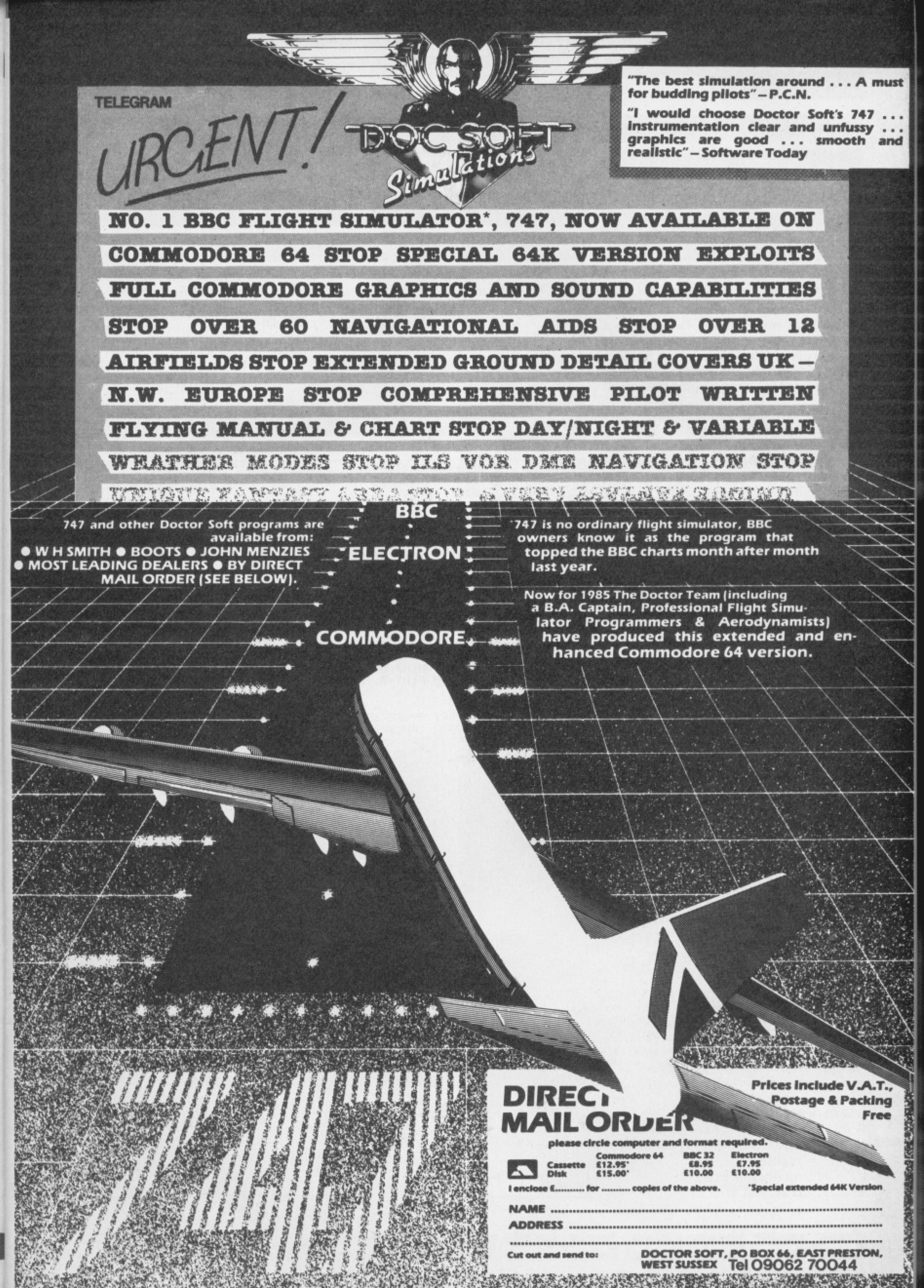
SYS(Location) targetline OR SYS Location (targetline)

The version as it stands is the coding for the Commodore 64. Lines 4100-4210 indicate the three changes that have to be made for the routine to work on a VIC. Lines 4200-4210 give the version for PETS (BASIC 2) and lines 4400-4410 give the version for PETS (BASIC 4). Naturally, as one would expect, this version is the fastest of all at a mere 0.02 seconds to search through 2000 data statements.

Preferences

I prefer the first version as the number of programs with 2000 data items in them is limited and even this case takes less than 10 seconds. But if you like speed at all costs then the machine code is obviously unbeatable.

```
B*
   PC   SR AC XR YR SP
.;0008 31 22 22 C1 F6
.
033C 20 8A AD  JSR $AD8A
033F 20 F7 B7  JSR $B7F7
0342 84 5F     STY $5F
0344 85 60     STA $60
0346 20 13 A6  JSR $A613
0349 A4 5F     LDY $5F
034B D0 01     BNE $034E
034D CA        DEX
034E 88        DEY
034F 84 41     STY $41
0351 86 42     STX $42
0353 60        RTS
```


TELEGRAM
URGENT!
DOC SOFT
Simulations
"The best simulation around . . . A must for budding pilots" – P.C.N.
"I would choose Doctor Soft's 747 . . . instrumentation clear and unfussy . . . graphics are good . . . smooth and realistic" – Software Today
NO. 1 BBC FLIGHT SIMULATOR*, 747, NOW AVAILABLE ON
COMMODORE 64 STOP SPECIAL 64K VERSION EXPLOITS
FULL COMMODORE GRAPHICS AND SOUND CAPABILITIES
STOP OVER 60 NAVIGATIONAL AIDS STOP OVER 12
AIRFIELDS STOP EXTENDED GROUND DETAIL COVERS UK –
N.W. EUROPE STOP COMPREHENSIVE PILOT WRITTEN
FLYING MANUAL & CHART STOP DAY/NIGHT & VARIABLE
WEATHER MODES STOP ILS VOR DME NAVIGATION STOP
BBC
ELECTRON
COMMODORE
747 and other Doctor Soft programs are available from:
• W H SMITH • BOOTS • JOHN MENZIES • MOST LEADING DEALERS • BY DIRECT MAIL ORDER (SEE BELOW).
*747 is no ordinary flight simulator, BBC owners know it as the program that topped the BBC charts month after month last year.
Now for 1985 The Doctor Team (including a B.A. Captain, Professional Flight Simulator Programmers & Aerodynamists) have produced this extended and enhanced Commodore 64 version.
747
DIRECT MAIL ORDER
Prices include V.A.T., Postage & Packing Free
please circle computer and format required.
Commodore 64 | BBC 32 | Electron
Cassette £12.95* | £8.95 | £7.95
Disk £15.00* | £10.00 | £10.00
I enclose £.......... for .... copies of the above.
*Special extended 64K Version
NAME ....
ADDRESS ....
Cut out and send to:
DOCTOR SOFT, PO BOX 66, EAST PRESTON, WEST SUSSEX Tel 09062 70044

In the final part of this series, David Rees has compiled an example game from the routines used in previous installments.

GAMESMANSHIP

TO CONCLUDE THIS SERIES, I HAVE chosen Asteroids, the old arcade favourite, as my example. The game is a fast version of the original and includes full instructions and a high score feature. It incorporates all the routines used in previous series apart from the scroll routine.

There are no embellishing stars behind the asteroids (see the beginning of part 2) because border characters are used to stop your laser bolt: extra stars would detonate your bolt prematurely. The score routine is the same as that used in part 3. The special effects used in the text pages illustrate good use of Extended Colour Mode, where only the first 64 characters of the set are available.

However, each screen position can have one of 4 chosen background colours. This lets you highlight text features and, as you may see from the display, adds interest to an otherwise static screen.

Happy zapping – you have a score of 2870 to beat!

How it works

10	set memory and key repeat
20	GOTO poke code and graphics
25-97	initialise registers
100-199	main routine
200-260	fire laser bolt
300-340	laser bolt range used up
350-370	sort out collision
380-420	you are hit, lose a life
450-498	asteroid hit by your laser bolt
500-580	POKE code into memory from DATA
600-660	POKE sprite DATA into memory
700-890	set machine code registers
900-990	set up display format
1000-2440	machine code DATA
3000-3200	sprite DATA
4000-4060	enter your name
4070-4120	file your score in the list
4200-4300	PRINT list and wait
4400-4450	another go?
4500-4750	PRINT out instructions and wait

Main Variables

G	direction the ship is pointing in
H,I,J	POKE positions for code registers
L	number of lives left
M	laser bolt fired or not
P	PEEK position for collision register
Q	PEEK position for laser bolt finish register
R1 to R4	routine starts
S	score
U	POKE position for code registers
V	start of video registers
NS	your name
A(7),B(7), C(7),D(7)	vectors for laser bolt move
S(9)	score list
NS(9)	names list

Program Listing

```
1 REM*****************
2 REM*  ASTEROIDS    *
3 REM*      BY       *
4 REM* DAVID REES    *
5 REM*    USING      *
6 REM*CODE ROUTINES*
7 REM*****************
10 POKE52,56:POKE56,56:POKE650,128
20 IFPEEK(49152)<>169THENGOSUB500
25 GOSUB4500:Q=53052
30 O=0:L=3:GOSUB700:S=0:M=0:P=780
40 POKE56325,100:T=π
45 U=53029:H=52994:I=53010:J=53041
50 FORN=0TO7
60 A=-INT(SIN(N*π/4+T)+0.5)*3:A(N)=ABS(A)
70 B=INT(COS(N*π/4+T)+0.5)*3:B(N)=ABS(B)
80 C(N)=SGN(A):D(N)=SGN(B):IFC(N)=-1THENC(N)=0
90 E(N)=A:F(N)=B:IFD(N)=-1THEND(N)=0
95 NEXT
97 G=0
100 SYSR1:SYSR2:IFPEEK(P)<>0THEN350
110 SYSR3:IFPEEK(Q)<>32THEN300
120 GETA$:IFA$=""THEN100
130 IFA$="/"ANDM=0THEN200
```



```
140 IFA$="."THENG=(G+1)AND7:GOTO170
150 IFA$<>","THEN100
160 G=(G-1)AND7
170 A=A(G):B=B(G):POKE2040,246+G:GOTO100
199 END
200 SYSR4:POKEH,A(G):POKEI,B(G)
210 POKEH+1,C(G):POKEI+1,D(G):M=1
220 POKEV+2,179+E(G)*4
230 POKEV+3,145+F(G)*4
240 FORN=0TO3:POKEU+N,1:NEXT
250 POKEV+16,PEEK(V+16)AND253
260 GOTO100
300 POKEV+2,20:POKEV+3,244
310 FORN=0TO3:POKEU+N,8:NEXT
320 POKEV+16,PEEK(V+16)AND253
330 POKEH,0:POKEI,0:M=0
340 GOTO100
350 FORN=0TO7:IFPEEK(J+N)<>0THEN370
355 NEXT
360 GOTO100
370 IFN>3THEN450
380 L=L-1:IFL<0THEN4000
390 O=1:GOSUB700:POKER4+26,20:SYSR4
400 FORN=0TO78:POKEV+39,N:NEXT
409 REM*[HOME][RVS ON][LEFT]
410 PRINT"■"SPC(37)"■"L"■ "
420 POKER4+26,50:M=0:GOTO100
450 N=N-3:POKEV+3+N*2,0
455 S=S+INT(PEEK(H+N*2)*PEEK(I+N*2))*5
460 POKEH+N*2,1+RND(1)*3
465 POKEI+N*2,1+RND(1)*3
470 POKEH+1+N*2,INT(RND(1)*2)
475 POKEI+1+N*2,INT(RND(1)*2)
479 REM*[HOME][RIGHT*6][RVS ON][LEFT]
480 PRINT"■","▶▶▶▶▶▶■"S"■ "
490 GOTO300
498 END
499 REM*CODE POKE*
500 A(1)=13139:A(2)=27770:A(3)=13025
510 A(4)=3477:B=48896
520 FORN=1TO6:T=0
530 FORM=0TO255:READA:IFA<0THEN550
540 POKEB+N*256+M,A:T=T+A:NEXT
549 REM*[DOWN]
550 PRINT"■TOTAL FOR ROUTINE"N" IS:"T
560 PRINT"IT SHOULD BE:"A(N)
570 IFA=-2THEN600
580 NEXTN
599 REM*SPRITE POKE*
600 B=15744
610 FOR N=0 TO 9
615 C=B+N*64
620 FORM=0TO63:READA:IFA=-1THEN640
630 POKEC+M,A:NEXT
640 FORP=MTO63:POKEC+P,0:NEXT
650 NEXTN
660 RETURN
699 REM*SET CODE REGISTERS*
700 POKE52992,0:POKE53008,0
710 POKE52994,0:POKE53010,0
720 FOR N=2 TO 5
730 POKE52992+N*2,1+RND(1)*3
740 POKE53008+N*2,1+RND(1)*3
750 POKE52993+N*2,INT(RND(1)*2)
760 POKE53009+N*2,INT(RND(1)*2)
770 NEXTN
780 FOR N=0 TO 7
790 POKE53025+N,((N/4)AND1)*8
800 POKE53033+N,(NAND3)+2
810 NEXTN
820 POKE53024,3
830 FOR N=1 TO 7
840 POKE53049+N*2,0
850 NEXTN
860 POKE53065,12:POKE53066,12
870 POKE53051,1:POKE2024,32
880 R1=49152:R2=49152+256
890 R3=49152+512:R4=49152+768
899 REM*VIDEO DISPLAY
900 V=53248:POKEV+32,0:POKEV+33,0
905 POKEV+17,PEEK(V+17)OR64
910 POKEV+16,0:POKEV+21,O*63:POKEV+27,255
915 POKEV,179:POKEV+1,145
920 POKEV+2,20:POKEV+3,244:X=1024
925 FOR N=2 TO 5
930 POKEV+N*2,RND(1)*255:POKEV+1+N*2,0
935 POKEV+39+N,5:POKE2040+N,255:NEXTN
937 IFO=1THENRETURN
940 POKEV+39,14:POKEV+40,1:POKEV+34,6
944 REM*[CLR HOME][YEL][L.BLUE]
945 PRINT"▔▝ASTEROIDS▔  SCORE: 0";
949 REM*[LEFT]
950 PRINT"          LIVES LEFT:"L"■  "
955 FORN=0TO39:POKEX+N,PEEK(X+N)+64:NEXT
960 FORN=1TO24:POKEX+N*40,160
965 POKEX+N*40+39,160:NEXT
970 X=1024+960:POKEV+35,6
975 FORN=0TO39:POKEX+N,160:NEXT
980 POKE2040,246:POKE2041,254
985 POKEV+23,63:POKEV+29,63
990 POKEV+21,63:RETURN
998 REM* ROUTINE 1 *
999 REM*SPRITE MOVE*
1000 DATA169,1,133,251,160,,173,16,208
1010 DATA37,251,240,2,169,1,133,252
1020 DATA185,,208,190,1,207,224,0
1030 DATA240,12,24,121,,207,144,3,24
1040 DATA230,252,24,144,9,56,249,,207
```



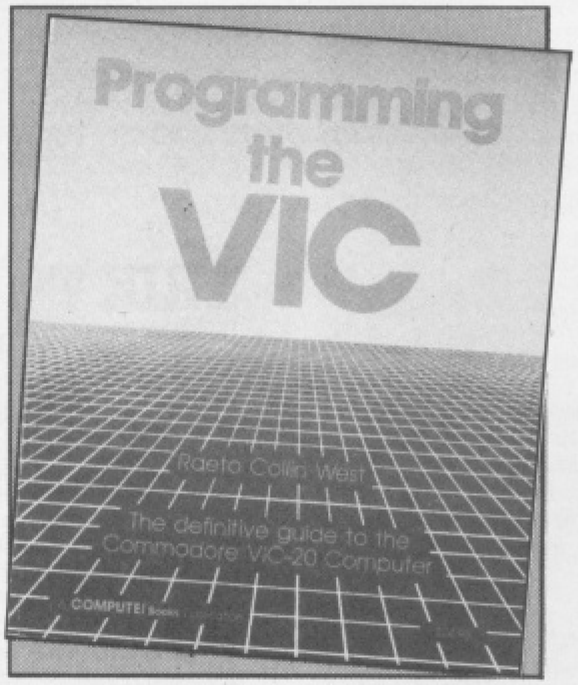
Program Listing (cont.)

```
1050 DATA176,2,198,252,24,153,,208
1060 DATA165,252,41,1,240,8,165,251
1070 DATA13,16,208,24,144,7,165,251
1080 DATA73,255,45,16,208,141,16,208
1090 DATA185,1,208,190,17,207,224,0
1100 DATA240,7,24,121,16,207,24,144,4
1110 DATA56,249,16,207,24,153,1,208
1120 DATA6,251,200,200,192,16,208,154
1130 DATA96,-1
1198 REM*      ROUTINE 2         *
1199 REM*SPRITE/SPRITE COLLISION*
1200 DATA173,32,207,41,7,141,32,207
1210 DATA160,,132,2,185,33,207,201,8
1220 DATA240,111,24,10,24,168,24,74,24
1230 DATA170,169,1,224,,240,6,24,10,24
1240 DATA202,208,250,45,16,208
1250 DATA240,2,169,128,133,251,185,,208
1260 DATA24,74,24,101,251,24
1270 DATA141,250,207,185,1,208,24,74,24
1280 DATA141,251,207
1290 DATA164,2,185,41,207
1300 DATA24,10,24,168,24,74,24
1310 DATA170,169,1,224,,240,6,24,10,24
1320 DATA202,208,250,45,16,208
1330 DATA240,2,169,128,133,251,185,,208
1340 DATA24,74,24,101,251,24
1350 DATA141,252,207,185,1,208,24,74,24
1360 DATA141,253,207
1370 DATA24,144,6,24,144,136,24,144,77
1380 DATA173,250,207,56,237,252,207
1390 DATA176,2,73,255,24,172,32,207
1400 DATA192,,240,5,74,24,136,208,251
1410 DATA201,,240,5,169,,24,144,2
1420 DATA169,1,133,253,173,251,207
1430 DATA56,237,253,207,176,2,73,255
1440 DATA24,172,32,207,192,,240,5,74
1450 DATA24,236,208,251,201,,240,5,169,
1460 DATA24,144,2,169,1,37,253
1465 DATA24,144,2,169,,164,2
1470 DATA153,49,207,200,192,8,208,161
1480 DATA160,,169,,24,121,49,207,24,200
1490 DATA192,8,208,246,96,-1
1498 REM*      ROUTINE 3         *
1499 REM*SPRITE/BCKGRD COLLISION*
1500 DATA169,1,133,251,160,,132,2
1510 DATA185,57,207,240,81,173,16,208
1520 DATA37,251,240,2,169,128,133,252
1530 DATA185,,208,24,74,24,101,252,56
1540 DATA233,20,24,109,73,207,24,74,24
1550 DATA74,24,133,252,185,1,208,24,74
1560 DATA56,233,33,24,109,74,207,24,74
1570 DATA24,74,24,168,173,136,2,133,253
1580 DATA165,252,192,,240,11,24,105,40
1590 DATA144,3,24,230,253,136,208,245
1600 DATA133,252,160,,177,252,164,2
1610 DATA153,58,207,6,251,200,200,132,2
1620 DATA192,16,208,157,96,-1
2398 REM* ROUTINE 4 *
2399 REM* GUN SHOT  *
2400 DATA169,,141,4,212,169,10
2410 DATA141,5,212,169,1,141,6,212
2420 DATA169,15,141,24,212,169,129
2430 DATA141,4,212,169,50,141,1,212
2440 DATA96,-2
2999 REM*SPRITE DATA*
3000 DATA12,,,12,,,18,,,18,,,33,,,33,,
3010 DATA64,128,,64,128,,63,-1
3020 DATA,,,1,128,,6,128,,57,,,65,,
3030 DATA66,,,34,,,18,,,12,-1
3040 DATA,,,48,,,88,,,67,,,64,192,,67,,
3050 DATA88,,,48,-1
3060 DATA,,,12,,,18,,,34,,,66,,,65,,
3070 DATA57,,,6,128,,1,128,-1
3080 DATA,,,63,,,64,128,,64,128,,33,,
3090 DATA33,,,18,,,18,,,12,,,12,-1
3100 DATA,,,12,,,18,,,17,,,16,128,
3110 DATA32,128,,39,,,88,,,96,-1
3120 DATA,,,3,,,12,128,,48,128,
3130 DATA192,128,,192,128,,48,128,
3140 DATA12,128,,3,-1
3150 DATA96,,,88,,,39,,,32,128,
3160 DATA16,128,,17,,,18,,,12,-1
3170 DATA,,,,,,,,,12,,,30,,,30,,,12,-1
3180 DATA60,,,195,,,128,128,,128,128,
3190 DATA65,,,33,,,65,,,128,128,,205,,
3200 DATA114,-1
4000 FORN=0TO29:GETA$:NEXT
4009 REM*[CLR HOME]
4010 POKE56325,50:PRINT"]";:POKEV+21,0
4020 IFS(9)>=STHEN4200
4030 PRINT"YOU ARE IN THE TOP TEN SCORES"
4039 REM*[DOWN*2]
4040 PRINT"■■PLEASE ENTER YOUR NAME:"
4050 INPUTN$
4060 N$=LEFT$(N$,14)
4070 N=0
4080 FORN=0TO9:IFS>S(N)THEN4100
4090 NEXT
4100 FORM=9TONSTEP-1
4110 N$(M+1)=N$(M):S(M+1)=S(M):NEXT
4120 S(N)=S:N$(N)=N$
4199 REM*[CLR HOME]
4200 PRINT"]";
4209 REM*[RVS ON][L.BLUE]
4210 PRINTSPC(15)"■■ASTEROIDS"
4219 REM*[PURPLE][DOWN]
4220 PRINTSPC(16)"■SCORES■"
4229 REM*[WHITE]
4230 PRINT" ■1 "N$(0)
4239 REM*[UP][DOWN][GREEN]
4240 PRINT"]"TAB(20);S(0)"■■"
4250 FORN=1TO9
4260 PRINTN+1;N$(N)
4269 REM*[UP][DOWN]
4270 PRINT"]"TAB(20);S(N)"■"
4280 NEXT
4289 REM*[RVS ON][L.BLUE]
4290 PRINT"■■PRESS ANY KEY TO CONTINUE"
4300 GETA$:IFA$=""THEN4300
4399 REM*[CLR HOME]
4400 PRINT"]"
4410 PRINT"WOULD YOU LIKE ANOTHER GO?"
4420 PRINT"ANSWER 'Y' OR 'N'"
4430 GETA$:IFA$="Y"THEN25
4440 IFA$<>"N"THEN4430
4450 END
4499 REM*INSTRUCTIONS*
4500 V=53248:POKEV+32,0:POKEV+33,0
4505 POKEV+17,PEEK(V+17)OR64
4509 REM*[CLR HOME][RVS ON][L.BLUE]
4510 PRINT"]"SPC(16)"■■ASTEROIDS"
4519 REM*[YELLOW]
4520 PRINTSPC(15)"■INSTRUCTIONS"
4529 REM*[DOWN*2][GREEN]
4530 PRINT"■■■ YOU ARE IN AN";
4540 PRINT" ASTEROID FIELD TRAPPED";
4550 PRINT" INA SPACESHIP ";
4560 PRINT"WITH CRIPPLED MAIN MOTORS.";
4570 PRINT"  ALL YOU HAVE TO ";
4580 PRINT"MANOEUVRE WITH ARE";
4590 PRINT"        GUIDANCE JETS."
4600 PRINT" YOU HAVE TO DEFEND YOURSELF";
4610 PRINT" FOR AS LONGAS POSSIBLE";
4620 PRINT" AGAINST THE JAGGED ASTEROIDS";
4630 PRINT"USING YOUR SKILL AND LASER";
4640 PRINT" CANNON."
4649 REM*[L.BLUE][RVS ON]
4650 PRINT" ■■GOOD LUCK!"
4659 REM*[DOWN*2][GREEN]
4660 PRINT"■■■KEYS ARE:"
4669 REM*[L.BLUE]
4670 PRINT"■'<' AND '>' TO TURN";
4680 PRINT" THE SHIP."
4690 PRINT"'/' TO FIRE THE LASER."
4699 REM*[DOWN*2][RVS ON]
4700 PRINT"■■■PRESS ANY KEY TO CONTINUE"
4710 POKEV+35,2:GETA$
4720 FORN=0TO199:NEXT
4730 POKEV+35,6:FORN=0TO199:NEXT
4740 IFA$=""THEN4710
4750 RETURN
```

REFERENCE

Garry Marshall assesses Raeto West's book on 'Programming the VIC' and Allen and Margaret Webb look at a selection of other Commodore books.

Title: 'Programming the VIC'
Author: Raeto West
Publisher: Compute! Publications
Price: £15.90



IF EVER THERE WAS A BOOK THAT TRIED to be all things to all VIC users, this is it. It deals with the architecture of the VIC, BASIC programming and machine code programming. Then it takes the reader beyond VIC-BASIC, shows how BASIC and machine code can be mixed and delves into the machine code routines that are contained in the VIC's ROM. After this it examines the creation of sound and graphics before describing the peripherals that can be attached.

This brief synopsis should serve to show that there are very few aspects of using the VIC that are not dealt with. The book is aimed at all levels of users from the beginner to the expert, and there are parts that are aimed at the absolute beginner as well as programming tips that will benefit the most advanced expert.

Given its blanket coverage of VIC programming and its claim to the widest possible readership, it must be said first of all that the book is a success. It is hard to imagine that more information on the VIC could be covered and packed into a single book. The book does cater for all VIC owners. There are a few shortcomings, but they are really rather minor in light of the book's overall achievement.

To try to be a little more specific in assessing the book, we can start by looking a little more closely at its contents, and then try to see from this how the treatment is geared to various classes of VIC users.

The first chapter is aimed at the absolute beginner, and does little more than describe what you can see when you look at a VIC. I got the strong impression that it was included more from a sense of duty than anything else. The chapter is needed to give balance to the book, but I don't think that the author's heart was in it: he was much more interested in getting down to the 'nitty gritty' of looking inside the VIC and stretching its capabilities than in introducing it. To support this idea, consider that this chapter has five pages, whereas the one that tucks into advanced BASIC programming has 74.

Preliminaries over, we are plunged into a BASIC reference guide. This gives a comprehensive coverage of the rules of VIC-BASIC, of all its keywords (each of which is discussed and illustrated with an example), and of the error messages that are given when something goes wrong with a program. This shows one of the book's strengths, for it is an ideal reference guide for the VIC user. And this is not only true for BASIC, for later on we get an equally good and complete guide to 6502 machine code and a further one to the routines in the VIC ROM and their locations. The fairly experienced programmer who writes programs in BASIC, in machine code, and in a mixture of both, and who wants to call on the routines that the VIC contains to save unnecessarily rewriting what is already under his nose need look no further for the perfect source of reference.

After the BASIC reference guide comes an introduction to BASIC programming. I was a bit uneasy about this chapter, feeling that the author was still straining to get to the parts that really interested him. It isn't clear to me at whom the chapter is aimed, as it seems to fall between what beginners and experienced programmers would need. It is too general and unstructured for the former and rather unnecessary for the latter. It does at least contain plenty of example programs illustrating most of the features of BASIC. And, in a nice touch, as soon as it gets to a program of any length it introduces a checking program that will help the inexperienced user to ensure that programs are typed correctly.

With chapter 5, and by now we are a hundred pages in, the book really starts to hum. This is where the author wanted to start. We can forgive a small 'faux pas' on its first page ('A bit, or binary digit, is a single, tiny electronic switch which can be either on or off') as the result of his enthusiasm to tell us about the VIC's chips, memory maps, unexpanded and expanded configurations, interfaces and more. The first program in this chapter is particularly impressive in that it allows us to 'see' the hardware in action. It puts a machine code routine in the cassette buffer which, when run with the appropriate parameters, shows what is going on in such places as the input buffer and the screen memory. This is really ingenious, and overcomes the problem that a great many people have in understanding hardware because they can't see anything happening.

'Beyond VIC-BASIC' successfully guides us just there. It explains how BASIC programs are stored. This opens up an otherwise incomprehensible area by allowing us to write programs that can manipulate and change other programs. The importance of this need not be doubted for long when you consider that the interpreters and compilers that are essential to all high-level programming do just this. They change programs by translating them to machine code and, of course, the programs cannot be run at all unless this is done. Among other things, this chapter also presents a number of programs for very useful toolkit routines such as the ones for merging programs and reconfiguring the VIC for some special purpose.

The next few chapters deal with machine code, and after a general introduction to 6502 machine code, methods specific to the VIC are examined. It is a good idea to tackle machine code in a way that takes advantage of what the VIC can do, and some very short programs can produce quite remarkable results. But I did feel that the treatment shared some of the shortcomings of the introduction to BASIC in that it fell between the same two stools in its approach and that it rather lacked a sense of direction. There are plenty of examples again, but I felt that they seemed more like a collection of isolated effects rather than seeming to build towards a coherent whole.

Following this there are 82 pages on graphics and ten on sound. This is a little unfair to the sound, which is quite as attractive a feature as graphics, but on the other hand the treatment of the graphics is pretty good. Apart from explaining about scrolling, panning and animation, it explains how to do something about that

LIBRARY

funny-sized screen. If you didn't expect to find out how to do something about the screen from the chapter on graphics, it is worth mentioning that the book has a good index, and that this not only makes it easy to trace any of the information contained in the book but also increases its value as a source of reference.

To try and sum up, I would say that the book could scarcely be bettered as a reference guide for the VIC, and as an aid and source of ideas to the more advanced programmer, whether using BASIC or machine code. I wouldn't recommend it to the beginner, though. For anyone in between the novice and advanced stages of VIC programming, this is still probably as good a buy as any single book that is available. But, to illustrate that there are problems with it for the intermediate user, consider this example from page 94. The BASIC lines,

```
100 FOR J=0 TO -1 STEP 0
110 ...
120 J=(A=B)
130 NEXT
```

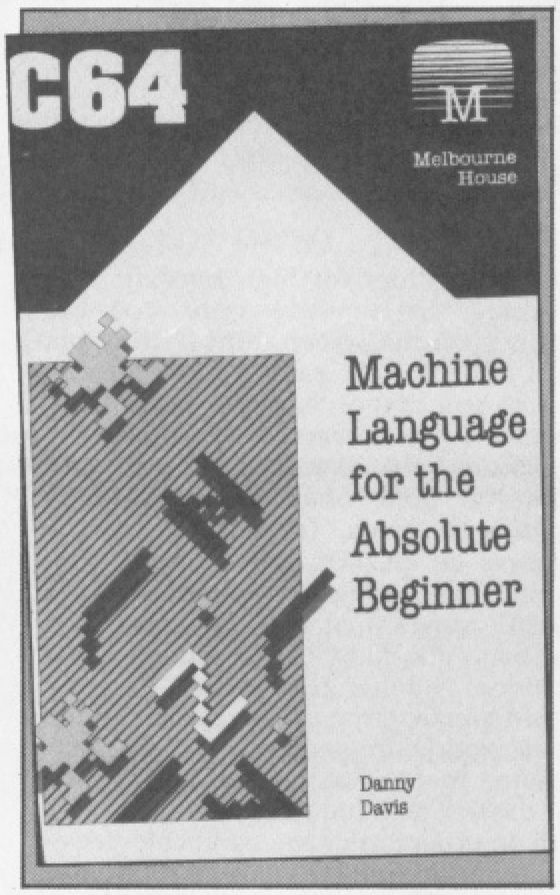
have the same effect as REPEAT ... UNTIL A=B. This is ingenious, and could provide a useful idea for anyone. But I think that a great many people might come back a week later to a program incorporating this trick and wonder how on earth it works!

I also feel that it is a shame that the book didn't appear rather earlier.

Title: Machine Language for the Absolute Beginner
Author: Danny Davis
Publisher: Melbourne House
Price: £7.95

MACHINE LANGUAGE PRESENTS AN interesting paradox. On the one hand it is extremely fast, efficient and compact. On the other hand, it is a language which can be difficult to learn, particularly in the absence of good text books. With every youngster wanting to be an arcade game writer, it is hardly surprising that a variety of books have appeared on the subject of machine code.

This book, published by Melbourne House, is an A5 size volume of 194 pages aimed at the absolute beginner. Much to his credit, the author has identified the problem that to make learning machine code easier, it is vital to have an easy method of entering and running simple routines. To this end, a BASIC utility called ALPHA is included in the book. With this program you can assemble code,



disassemble, save and load code from cassette, and run routines. A simple facility is also included to allow the observation of a specified memory location during the execution of your machine code.

One disappointing aspect of this book is that only the first 100 or so pages are directly related to teaching machine code. The remainder contains 14 appendices which cover areas such as the memory map, 6510 instruction codes, details of the VIC and SID chips and a listing of ALPHA. Apart from ALPHA most of this seems to be padding.

The format of each chapter is interesting in that a summary of what has been discussed is given at the end. This helps the revision of the material studied. These summaries can be used as quick checks if you have a need to refer to the book in the future.

The material covered is much as expected in this sort of book, and doesn't offer any surprises. To assist the conversion from BASIC to machine code, many short BASIC routines are given to illustrate points. A comparable number of example machine code routines are given for you to input via ALPHA.

Overall this is a competent book which presents the material in a tolerably clear manner with a reasonable number of useful examples. The inclusion of a simple assembler is a real bonus which certainly makes this book worth buying.

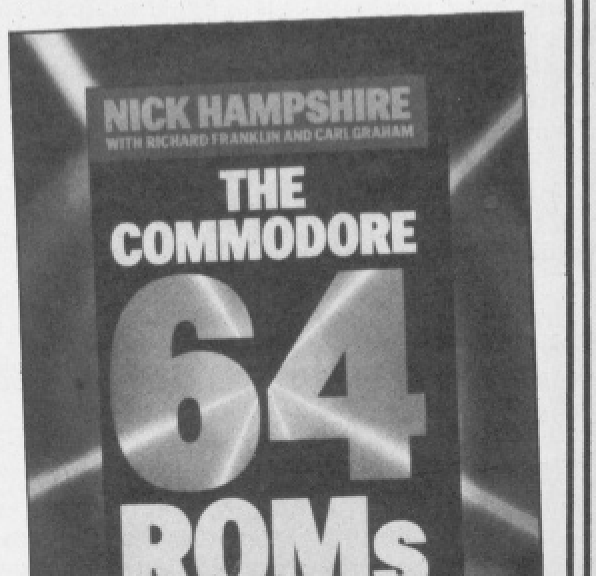
Title: The Commodore 64 ROMs Revealed
Author: Nick Hampshire, Richard Franklin and Carl Graham
Publisher: Collins
Price: £8.95

NEXT TO A GOOD QUALITY assembler, the most valuable tool in a serious programmer's armoury is a detailed disassembly of the computer's ROMs and a full list of the important entry points. The reason for this is simple. Inside the ROMs is a huge library of useful routines which can save a lot of time and irritation if you know how to use them.

Most PET users will remember Nick Hampshire's excellent volume "The PET revealed". Now he and his co-workers have applied themselves to assisting 64 users. This book is up to the same high standard as his PET volume.

It should be appreciated that there is more than one ROM disassembly on the market. So far, all of them have been bare listings with no labels and no helpful comments. Whilst being better than nothing, such documents are difficult to use and tedious to examine. Mr Hampshire has anticipated this problem and produced a highly readable work. In effect, a source code has been recreated from the object code.

The listing is fully annotated with labelled variables and loops. As far as possible, Commodore's own labels have been used. Each important routine is described in REMs and the entry conditions described. According to the forward, the code was finally tested by reassembling and comparing to the ROM contents. This, as a side effect, threw up the changes given in the new ROMs. Overall this must have been a monumental task requiring great patience.



REFERENCE LIBRARY

The source code, all 203 pages of it, is supplemented by a fair amount of additional information. This includes details of the ROM update, a detailed memory map and a full list of the important entry points.

This is a highly detailed and useful reference work which is essential reading to any serious 64 owner who wants to get the best from his machine. I recommend it.

Title: Supercharge Your Commodore 64
Author: Barry Thomas
Publisher: Melbourne House
Price: £6.95

MOST ARCADE GAMES ADVERTS INSIST on thrusting down our throats that machine code is fast and efficient and BASIC is slow and inefficient. This certainly holds true for arcade games but it's not entirely true. For most applications BASIC will do perfectly well, although some slow sections may require a byte or two of machine code. This book is intended to supply a library of ready made routines for such a purpose. All routines are supplied as a BASIC loader, a flow sheet and the source code. In keeping with most works of this type, the areas tackled are graphics, sprites, sound and utilities. As such there are some interesting inclusions and some remarkable exclusions.

First, let's look at sound. Unfortunately, this is the weakest section in the book. In fact, all you get are seven sound effect routines. Whilst this will be welcome to those of you who want to have a gun shot or explosion in your program, it's no use at all if you want to play a tune. It wouldn't have been beyond the wit of man to include some useful commands such as envelope etc.

The sprite routines proved to be both interesting and irritating. There are routines for turning on sprites, changing colours (hi-res only) reversing, inverting and erasing designs. The author, has for some obscure reason however, ignored the obvious aspects of sprite positioning, multicolour mode, expansion, collisions and priorities.

The routines for high resolution bit mapping are similarly weak. You can really clear the screen, turn it on or off, and move the bit pattern from place to place. You cannot, however, set a point, draw a line or do any drawing of any sort.

Finally, what about the utilities? Again rather a poor bunch. A RENUMBER routine is given, for example, which ignores all GOTOs and GOSUBS. This would be useless for renumbering a BASIC program of any significant size. Probably the most useful routine is the Random Number generator.

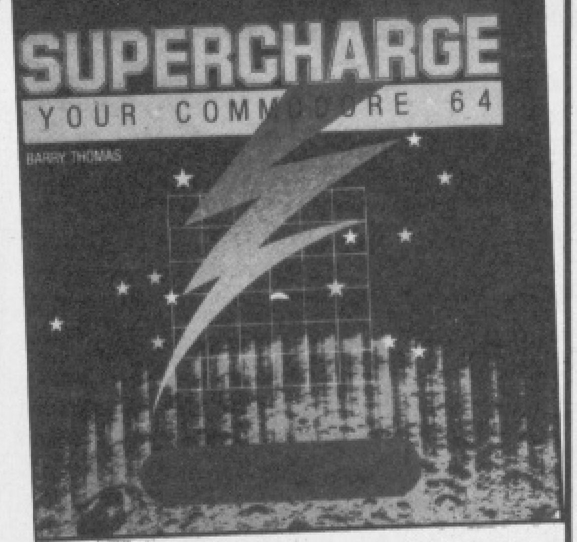
In simple terms, this is a case of a good idea poorly implemented. I can only assume that the author was constrained by the size of routine included since there are no programs of any significant power or sophistication. My advice to anyone thinking of buying this book is, forget it, it's a better idea to buy a teach yourself machine code book and do the job yourself.

Title: The Commodore 64 for kids of all ages
Author: Tony Noble
Publisher: Sigma Press
Price: £6.95

WITH SO MANY CHILDREN GETTING computers for birthdays, Christmas or simply as a toy for father, it isn't surprising that books are being published with the sole intention of teaching kids to write programs. This book assumes no knowledge of the 64 and teaches the rudiments of programming in easy steps. To give you something concrete to do when you've learned all there is to learn, there are some games to type in at the end of the book.

To make matters simple for the student, the author has adopted a neat and clear approach to displaying instructions. Key presses, for example, are depicted by large graphical representations of the keys. Rather than give large chapters containing indigestible lumps of information, the information is supplied in small steps each a page or two in length. Each step tackles a new aspect and gives detailed instructions on how to enter an exercise and, above all tells you what to expect on the screen.

Melbourne House

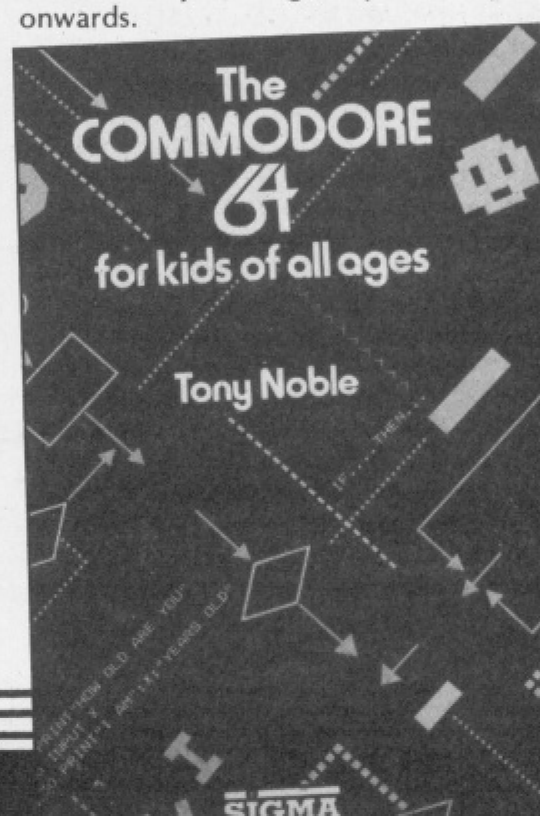


So what material is covered? Well, the answer is almost everything. The book starts by getting the user aquainted with using the keyboard and progresses through putting information on the screen by using print commands. It then moves on to colour and the standard graphics.

The remainder of the book moves into the realms of programming in BASIC and does so in the same gentle small increments. Each aspect of programming is illustrated clearly with practical examples. Even the more advanced aspects of sound, sprites, animation and randomness are dealt with effectively.

Once you've grasped the rudiments of BASIC, the design of programs is discussed with reference to the programs given in the book. Finally, there is a variety of programs covering education (spelling and arithmetic), adventure type games and logic.

This is an excellent book which tackles the tricky area of programming in a clear and entertaining manner. At over 200 pages it presents a huge volume of information and represents excellent value for money. This book is suitable for children of junior age, say from 7 years onwards.



SINGLE KEY ENTRY



Why press several keys when one will do? Faster programming is the goal of this utility from Les Allan.

THIS UTILITY, WHEN COMPLETED, gives the user the ability to enter KEY words with just a single key when used in association with the SHIFT key. For example SHIFT-A prints ASC, SHIFT-B prints STEP, SHIFT-C prints CHR$ and so on which will make typing in of programs much faster. A second set of keywords is also available so that it further increases the flexibility of the utility.

The program as listed must be entered as written and saved prior to running in case a typing error causes a system crash. Error trap routines have been incorporated so as to minimise that possibility. When completed insert a blank tape into the cassette, run the program and when prompted "HAVE YOU SAVED THIS PROGRAM Y/N" press Y. The machine code files SINGLE KEY ENTRY and SINGLE KEY will now be saved to tape. Provision has been made in the boot program such that this utility can be saved to tape or disc without any changes.

Upon subsequent use simply type LOAD for tape or LOAD"SINGLE KEY ENTRY",8,1. for disc and the saved program will automatically load and set up the operating system to accept the new commands.

KEYWORD SET 1
(SYS 50000)

The following keywords are initially made available and can be switched in and out with SYS 50000.

A-ASC	B-STEP	C-CHR$	D-DIM	E-END	F-FOR	G-GET	H-STEP
I-INPUT	J-GOTO	K-GOSUB	L-LEFT$	M-MID$	N-NEXT	O-OR	P-POKE
Q-PEEK	R-RIGHT$	S-STR$	T-TAB(	U-USR	V-VAL	W-DATA	X-READ
Y-RESTORE	Z-SYS						

KEYWORD SET 2
(SYS 50400)

The following keywords can be accessed and switched in and out with SYS 50400

A-ABS	B-ATN	C-CLOSE	D-CLR	E-EXP	F-FRE	G-GET	H-FN
I-INPUT #	J-COS	K-LOAD	L-LIST	M-LOG	N-NEW	O-OPEN	
P-PRINT #	Q-RND	R-RND	S-SIN	T-TAN	U-SQR	V-VERIFY	W-WAIT
X-RETURN	Y-CMD	Z-AND					

The machine code program checks location $D4 and if the editor is in quote mode the normal graphic character is printed instead of the keyword. Typing either SYS 50000 or SYS 50400 will activate or de-activate each keyword set. Also hitting RUN/STOP and RESTORE will de-activate the routine.

Program Listing

```
10 REM ***** SINGLE KEY ENTRY FOR COMMODORE 64 *****
15 :
20 POKE53280,3:POKE53281,3
25 PRINTCHR$(147)CHR$(31)SPC(7)"**** SINGLE KEY ENTRY ****"
30 PRINT:PRINT
35 PRINT"THIS MACHINE CODE PROGRAM ALLOWS THE USE"
40 PRINT"OF  SINGLE KEY ENTRY  DIRECTLY FROM  THE"
45 PRINT"KEYBOARD WHEN USED IN  CONJUNCTION  WITH"
50 PRINT"THE SHIFT KEY. TWO COMPLETELY  DIFFERENT"
55 PRINT"SETS ARE PROVIDED WHICH ARE ACTIVATED OR"
60 PRINT"DE-ACTIVATED BY  THE USE OF THIS COMMAND"
65 PRINT"            SYS 50000      SET 1        "
70 PRINT"            SYS 50400      SET 2        "
75 PRINT"THIS PROGRAM INCLUDES A  M/C  ROUTINE TO"
80 PRINT"SAVE BOTH THE AUTO BOOT AND MAIN ROUTINE"
85 :
90 REM **************** HEX  LOADER ****************
95 :
100 INC=0:SUM=0
105 READDA$:IFDA$="END"THEN165
110 IFLEN(DA$)<>2ANDDA$<>"END"THEN205
115 H=ASC(LEFT$(DA$,1)):H1=(H-48)*16:IFH>57THENH1=(H-55)*16
120 H=ASC(RIGHT$(DA$,1)):H2=(H-48):IFH>57THENH2=(H-55)
125 BCD=H1+H2:IFBCD<0ORBCD>255THEN205
130 POKE50000+INC,BCD:INC=INC+1:SUM=SUM+BCD
135 PRINT:PRINTSPC(5)"DATUM LEFT FOR TRANSFER :";
140 PRINT624-INCCHR$(157)CHR$(32)CHR$(145)CHR$(145)
```

Program Listing (cont.)

```
145 GOTO105
150 :
155 REM *************** SAVE  ROUTINE ***************
160 :
165 IFINC<>624ORSUM<>85252THEN205
170 PRINTCHR$(147)"DATA TRANSFER COMPLETE"
175 PRINT:PRINT:PRINT"HAVE YOU SAVED THIS PROGRAM Y/N"
180 GETKEY$:IFKEY$<>"Y"ANDKEY$<>"N"THEN180
185 IFKEY$="Y"THENSYS50528
190 PRINTCHR$(17)CHR$(17)"SAVE"CHR$(34)"SINGLE KEY BOOT"CHR$(34)
195 PRINTCHR$(145)CHR$(145)CHR$(145)CHR$(145)
200 END
205 PRINTCHR$(147)"ERROR IN DATA STATEMENTS !!!":STOP
210 :
215 REM ******** HEX DATA FOR PROGRAM LISTER ********
220 :
225 DATA 20,2E,C4,AD,14,03,C9,7C,F0,14,8D,04,CE,AD,15,03
230 DATA 8D,05,CE,A9,7C,8D,14,03,A9,C3,8D,15,03,60,AD,04
235 DATA CE,8D,14,03,AD,05,CE,8D,15,03,60,EA,48,8A,48,98
240 DATA 48,A5,D7,48,A5,D4,F0,04,68,4C,E3,C3,68,C9,C1,90
245 DATA 52,C9,DB,B0,4E,38,E9,C1,AA,BD,EB,C3,A2,00,86,C6
250 DATA AA,A0,9E,84,22,A0,A0,84,23,A0,00,0A,F0,10,CA,10
255 DATA 0C,E6,22,D0,02,E6,23,B1,22,10,F6,30,F1,C8,B1,22
260 DATA 30,11,08,8E,FF,CD,E6,C6,A6,C6,9D,77,02,AE,FF,CD
265 DATA 28,D0,EA,E6,C6,A6,C6,29,7F,9D,77,02,A9,14,8D,77
270 DATA 02,E6,C6,68,A8,68,AA,68,4C,31,EA,C6,A9,C7,86,80
275 DATA 81,A1,90,85,89,8D,C8,CA,82,9F,97,C2,C9,C4,A3,B7
280 DATA C5,83,87,8C,9E,7F,00,00,00,FF,EE,EE,EE,EE,EE,EE
285 DATA A9,0F,8D,20,D0,A9,06,8D,21,D0,A9,01,8D,86,02,A2
290 DATA 00,BD,40,C4,F0,07,20,D2,FF,E8,4C,21,C4,60,20,10
295 DATA C4,A2,00,BD,68,C4,9D,EB,C3,E8,E0,20,D0,F5,60,00
300 DATA 93,12,20,53,49,4E,47,4C,45,20,4B,45,59,20,45,4E
305 DATA 54,52,59,20,42,59,20,4C,45,53,20,41,4C,4C,41,4E
310 DATA 20,9A,92,0D,0D,0D,00,C6,A9,C7,86,80,81,A1,90
315 DATA 85,89,8D,C8,CA,82,B0,97,C2,C9,C4,A3,B7,C5,83,87
320 DATA 8C,9E,7F,00,00,00,FF,EE,EE,EE,EE,EE,EE,EE,EE,EE
325 DATA EE,EE,EE,EE,EE,EE,EE,EE,B6,C1,A0,9C,BD,B8,A1,A5
330 DATA 84,BE,93,9B,BC,A2,9F,98,BB,8A,BF,C0,BA,95,92,8E
335 DATA 9D,AF,7F,00,00,00,FF,EE,EE,EE,EE,EE,EE,EE,EE,EE
340 DATA EE,EE,EE,EE,EE,EE,EE,EE,EE,EE,EE,EE,EE,EE,EE,EE
345 DATA EE,EE,EE,EE,EE,EE,EE,EE,EE,EE,EE,EE,EE,EE,EE,EE
350 DATA 20,10,C4,A2,00,BD,98,C4,9D,EB,C3,E8,E0,20,D0,F5
355 DATA 4C,53,C3,EE,EE,EE,EE,EE,D0,E4,EE,EE,EE,EE,EE,EE
360 DATA A9,00,8D,20,D0,8D,21,D0,A9,01,A6,BA,A0,FF,20,BA
365 DATA FF,A9,0A,A2,F0,A0,02,20,BD,FF,A9,00,A2,FF,A0,FF
370 DATA 20,D5,FF,A9,83,8D,02,03,A9,A4,8D,03,03,A9,0F,8D
375 DATA 20,D0,A9,06,8D,21,D0,20,50,C3,4C,94,E3,00,00,00
380 DATA 00,00,00,00,00,00,00,00,00,53,49,4E,47,4C,45,20
385 DATA 4B,45,59,00,00,00,00,00,00,8B,E3,A7,02,EE,EE,EE
390 DATA A2,00,BD,00,C5,9D,A7,02,E8,E0,5D,D0,F5,A9,01,A6
395 DATA BA,A0,01,20,BA,FF,A9,10,A2,43,A0,C4,20,BD,FF,A9
400 DATA A7,85,2B,A9,02,85,2C,A2,04,A0,03,A9,2B,20,D8,FF
405 DATA A9,01,A6,BA,A0,01,20,BA,FF,A9,0A,A2,43,A0,C4,20
410 DATA BD,FF,A9,50,85,2B,A9,C3,85,2C,A2,00,A0,C5,A9,2B
415 DATA 20,D8,FF,A9,01,85,2B,A9,08,85,2C,4C,CA,02,EE,EE
420 DATA END
425 :
430 :
435 ***************************************************
440 *                                                 *
445 *      SINGLE KEY ENTRY FOR COMMODORE 64          *
450 *                                                 *
455 *        UTILITY COMMANDS ARE AS FOLLOWS          *
460 *                                                 *
465 *        SYS 50000         ACTIVATE SET 1         *
470 *                                                 *
475 *        SYS 50400         ACTIVATE SET 2         *
480 *                                                 *
485 *      REPEAT SYS COMMAND TO DE-ACTIVATE          *
490 *                                                 *
495 *                                                 *
500 *      LES ALLAN           19TH NOV 1984          *
505 *                                                 *
510 ***************************************************
READY.
```

Filing is not always an exciting task but it's very often necessary. A.P. and D.J. Stephenson show you how your computer can bring order to chaos.

THE BASIC FACTS



IN THIS PART OF THE SERIES we shall look at *storing* and *retrieving* information, a role which comes naturally to any computer. In other words, we are concerned with the techniques employed for building a kind of high-tech filing cabinet. We don't really need a computer to store information. In fact, storing is the easy part – just bung every bit of paper, considered to be of the slightest use in the future, into a cupboard or dressing table drawer. As the drawer fills up, start using another one and so on.

The trouble starts, some days or weeks later, when you want to retrieve a particular bit of paper urgently. The ensuing scene can be quite frightening to the casual observer. Papers are flung into the air to the accompaniment of verbal obscenities and other sound effects. The floor soon becomes covered with gas bills, pawn tickets, threatening letters and similar documents peculiar to the modern suburban dwelling.

It is no streak of fate that the document you want normally turns up near the bottom of the last draw searched. According to theoretical physics, the cause lies in a mysterious and malignant factor called 'entropy' which is associated with disorder and, what is worse, is always increasing! It certainly seems to work when we store papers in drawers. The disorder appears to increase at a rate proportional to the number of times the drawers are accessed.

Of course, a much better way to store information is in office type files. Each file cover has a title which describes the general nature of its contents while each piece of paper within the file is kept in place by a short tag string for maintaining *sequential integrity* – the most recent paper on top. Although this is a vast improvement on the drawer method, the system tends to degenerate after the enthusiasm for order wears off. It is all right in an office where someone is employed exclusively to maintain the files in order but in the relaxed atmosphere of the home papers soon get pushed into the wrong file (the tag threading is soon abandoned because it is too fiddly). After a few months the system is little more than an up-market version of the drawer method. So, if you have bought a computer, why not use it as a filing system? You will need to enter the information at the keyboard and a certain amount of dedication and discipline is still needed to maintain the files in good order. Programs for handling the data can include operations which would take far too long to complete by normal methods. The data can be sorted, modified, deleted or re-arranged in a variety of ways and, most important of all, individual items within a file can be extracted almost immediately.

Cassette data files

Although the floppy disc is the natural medium for storing file information, it is safe to assume that the majority of readers who own a Commodore machine will stick with cassette tape units for some time. Although speed of access is important in business files, the comparative slow speed of tape units can be tolerated in the home. The term 'file' is often used in rather a loose way. For example, we speak of program 'files' but all we really mean is that we have saved the complete program on tape – we have called the program itself a 'file'. A data file is something quite different. You can't RUN a data file because all it contains is information required by a program. The program must be loaded first and must contain lines which start reading in the data file. Obviously, the data file must already be resting in the cassette unit and rewound to the correct position.

The OPEN statement

Even in conventional filing systems, you can't put a new slip of paper in a folder unless you open it first. Similarly, you must *close* the folder again before slipping it back into the filing cabinet. It is hardly surprising therefore that BASIC keywords exist which have equivalent action. That is to say,

we have the OPEN statement and the CLOSE statement. The format of the OPEN statement for tape files is a bit frightening at first:

```
OPEN file number, device, number, secondary address, file name
```

The meanings are as follows:

File number: this can be any number you like between 0 and 255. It is just in case you want to distinguish between two different files under the same collective file name.

Device number: since data files are kept on peripheral equipment the computer must know which one. The tape unit in the CBM 64 is always device number 1.

Secondary address: this number informs the computer what you want the peripheral to do. (The term secondary 'address' is a misleading term because it is not an address at all in the normal sense.) In the case of the tape unit, you have the following choice:

0 = read information from tape
1 = write information to tape
2 = write information to tape and place an end-of-tape marker at the end.

File name: this can be any name of your own choice providing the number of characters does not exceed 16.

All this may seem like a load of gobbledegook until you study the following examples:

Example:
OPEN 1,1,0, "ORGANISMS"
This opens file number 1 for reading data from tape.

Example:
OPEN 1,1,1,"PLANETS"
This opens file number 1 for writing data to tape.

Example:
OPEN 1,1,2,"WEAPONS"
This opens file number 1 for writing data to tape and appends an end-of-file marker.

Example:
OPEN 1,1,0,N$

This opens file number 1 for reading data from tape. The file name has been previously assigned to the variable N$.

Note that the first two parameters are the same in all the above examples. The first one, the file number, could



have been any other chosen number but it is unusual to take advantage of different file numbers under the same file name so, from now on, we shall always use 1 as the file number – it makes life easier. The second number will always be 1 when using tape so the only difference is in the third number and the file name. There are several *default* states allowed for simplifying an OPEN statement but it is a lazy way out and against the interests of good structure.

The OPEN format for Commodore machines is a bit on the weird side, so some readers may be interested in the underlying history. It all began with the first machine which Commodore launched called the PET 2001. It was, arguably, the first machine to capture public interest in this country way back in prehistoric times – the early Seventies, that is. The designers, rightly or wrongly, decided to use a special input/output bus which was achieving some fame as the new 'standard'. This was originally proposed by the famous firm of Hewlett Packard for standardising computer control of instrumentation. It later became known as the IEE 488 Bus Protocol. Apart from the introduction of the terms 'Device address' and 'Secondary address', the important innovation was the concept of *daisy chaining* all peripherals together on the same bunch of eight data wires with another bunch of eight for passing control signals, device addresses and secondary addresses. The system was brilliant but it is a matter of conjecture whether Commodore was, in the original instance, wise to choose such a bus system. It was never intended to be employed in home computer systems – it was over sophisticated for such a purpose. As events have turned out, hardly any other home computer machine used the IEE 488 bus so Commodore machines occupy a somewhat lonely, although not necessarily inferior, position in this respect.

The CLOSE statement

Once the file has been opened and data transference is complete, it is advisable to close it immediately, even if you have to open it again later on. This is common sense in ordinary office procedure as well as in computerised versions. The greatest menace in any office is the person who has a habit of taking a file out of the cabinet and leaving it lying around. The file has not been closed properly until it has been returned to the cabinet so



other people can us it again. The equivalent action in computer terms is the CLOSE statement. The format is:

CLOSE file number

Since we have decided to always use 1 as the file number, the statement becomes,

CLOSE 1

To emphasise the importance of closing a file after you have finished tranferring data, you would do well to remember that all tape transfers take place between an area in RAM known as the tape buffer. The buffer contents are transferred a block at a time and any bits left over after the last completed transfer will be left there unless the file is CLOSed. In other words, until you CLOSE a file, the data transfer between computer and tape is not complete and you will get read errors when you later try to retrieve the data.

The PRINT # statement

When a tape file is OPENed, the PRINT # statement is normally used for 'printing' data to tape. The simplified format is:

PRINT # file number,variable

The file number must be the same as the file number used in the OPEN statement. Since we have agreeed to always use a file number of 1, some examples would be:

```
PRINT # 1,N
```

```
PRINT # 1,A$
```

```
PRINT # 1,A$(F,R)
```

```
PRINT # 1,FS%:PRINT # 1,NF%:
PRINT # 1,L%
```

A word of warning here about using commas or semicolons to separate variables: they do not have the same effect as they would in an ordinary PRINT statement. If punctuation is to be printed to tape, you should use the appropriate ASCII characters instead. For example, CHR$(44) must be used to print the comma. The last of the four examples above shows a safe way of printing *separate* variables to tape; separate PRINT # statements are used.

The INPUT # statement

This is the mirror image of PRINT # and is used to read back characters from a data tape. Some examples:

```
INPUT #1,N
```

```
INPUT # 1,A$
```

```
INPUT # 1,A$(F,R)
```

```
INPUT # 1,FS%,NF%,L%
```

Unlike the PRINT # statement, commas can be used in the INPUT # statement to separate variables.

Simple test programs

To get the feel of data tape programming, key in Program 7.1 which should print the numbers 1 to 10 on tape. Don't RUN it yet.

```
Program 7.1
100 REM PRINTING NUMBERS
ON TAPE
110 N$="TEST"
120 OPEN 1,1,1,N$
130 FOR N=1 TO 10
140 PRINT#1,N
150 NEXT
160 CLOSE1
```

RUN it only after you have ensured a blank cassette (or one recorded with data no longer required) is in position and rewound. The program should then print the numbers on tape.

Program 7.2 is a simple program for reading back the numbers from tape. Again, make sure the data tape is rewound and ready before running Program 7.2

```
Program 7.2
100 REM READING NUMBERS
FROM TAPE
100 N$="TEST"
120 OPEN 1,1,0,N$
130 FORN=1 TO 10
140 INPUT  1,N
          50 PRINT N
150 NEXT
160 CLOSE1
```

Once you are satisfied that you can send and retrieve numbers from tape, run the programs again but alter the FOR loop in both of them to read 1 TO 1000, or even 1 TO 10000. This will take some time but will provide a good test for your tape unit – a worth while exercise before trusting it with important data. The slightest read-back error in a number will provoke a corresponding error message from the operating system.

Now we have seen how to incorporate data tapes into programs, it is time to delve into some of the jargon employed in computer filing systems. For example, we should know precisely the difference between a file, a record and a field within a record. We must also know the difference between an *ordinary field* and a *key field.*



Files

We shall define a file as a set of data items, all relating to the same subject, which can be held on tape (or disc) and accessable by a suitable program. Depending on our interests (and age) we might keep files on Butterflies, Birds, Football teams, Names and addresses, etc etc.

Records

If, for example, we have a file on birds, then all the information relating to the Green Tit will be treated as a separate record. So the file named "BIRDS" would contain separate records for each type of bird. It must be possible, of course, to extract any individual record from the file for examination at any time.



Fields

A record on the Green Tit would contain separate bits of information such as colour, size, habitat, shape of beak and diet. These are called fields.

The key field is the one which is used to uniquely identify a record. This, in the case of our file on birds, would be the bird's NAME. It would be absurd to choose COLOUR as the keyfield because lots of different birds have the same colour but, hopefully, not the same name. A record is quickly located within a file by searching for the required keyfield.

Field headings

When a file is initially created, one of the first tasks is to decide on the heading of each field. This information is an essential part of the file and must be stored together with the actual information under each heading. **Figure 7.1** shows example field headings of a file.

Field length

The number of characters allowed in each field is called the field length. The number of characters allowed can be decided when a file is first set up (created) or, more simply, fixed at some value, say, 18 characters maximum if memory space is at a premium.

Simple file organisation

There are many ways in which a file can be laid out (organised) but for tape store files, the main body of the file is conveniently stored in the form of a two-dimensional (*rectangular*) array. That is to say, in the form of 'horizontal' rows and 'vertical' columns in an array such as A$(F%,R%). F% is an integer variable representing a particular *field number* of a record and R% represents a particular *record number*. The fields could be numbered starting from field 1 upwards.



Figure 7.1. Field headings created for a file called "BIRDS". Remember to store the headings along with the information stored under them.

In fact, this would appear to be the natural method of numbering. However, this would be extremely inefficient because, when an array is DIMensioned, the interpreter reserves three bytes for each string descriptor in the array. (One byte for string length and two bytes for the string address). Thus, in a file of 100 records, we could waste 300 bytes by not using the 0 field element. In view of this, the keyfield should be field 0, the next, field 1 and so on.

On the other hand, records are best numbered from Record 1 onwards because it leaves the 0 slot for field headings. For example, if each record is to have three fields, the keyfield heading would be in A$(0,0), the second heading in A$(1,0) and the third heading in A$(2,0). **Figure 7.2** shows how fields and records are related in the array.

File length

The file length is the number of records it contains. Because an array has to be DIMensioned, an estimate of the maximum number of records must be made during the creation of a new file. As a warning against greed, we should point out that for tape, the entire file must be capable of fitting into RAM. We can't bring out one particular record from within a tape file. In other words, we can't have *random access* to a record unless the whole file is loaded into RAM.

Heading information

A tape data file must contain certain heading information in addition to the rectangular array holding the records and field headings. This information will include the following together with some suggested variable names:

Maximum file size (FS%). This is required for DIMensioning purposes.

Actual file length (FL%).

Number of fields (NF%).

Writing complete filing programs

The information given above is an outline of tape files. Next month's issue will show how to construct subroutines which can be strung together to build practical filing systems.

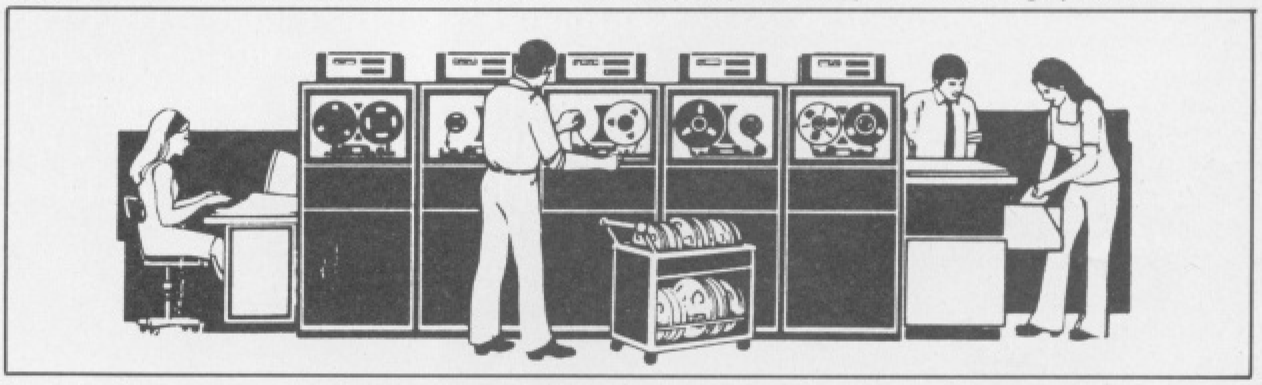
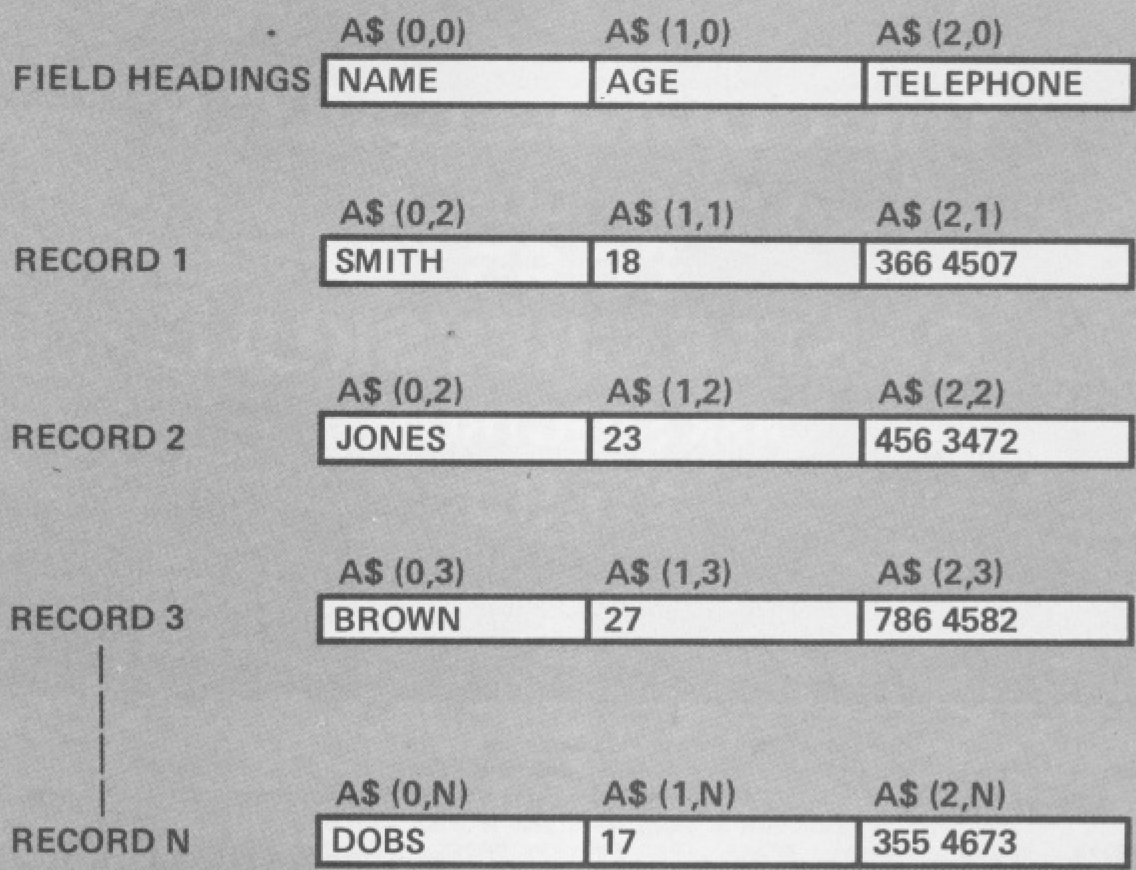


Figure 7.2. The records and field headings in this file are held in a 'rectangular' array. Each piece of information is given its own field number and record number





YEHTZEE

It all hangs on a throw of the dice in this computerised game by Jamie Clyde for up to six players.

YEHTZEE IS THE PERFECT example of a family or party game – this is one of a small number of games which allow more than one person to play at once. Up to six players can participate in this game which is a welcome change from the shoot'em-up type games of which there is already a myriad on the market nowadays.

The basic idea of the game is to fill your card by rolling dice and getting the required sequences. Once one of the combinations of the dice which you need turns up, you can mark it down on your card, winning the appropriate points. However, if you cannot do anything with your sequence, you must strike off one of the remaining choices When your card is complete the computer will total up the score and display it with your players' totals on the master scoreboard.

Playing the game

When the program is run, you will be greeted by a title page and a question: 'how many players?'. When you have entered the correct number, the screen will clear and you must enter the names of all the players, pressing RETURN after each one.

The game commences when the playing screen appears. The name of the player is displayed under the title and below that are the various combinations of the dice which must be sought. The top half of the card is made up of the individual numbers, for example 'threes' where you try to get as many threes as possible. This is exactly the same for the other numbers, except you must get as many of that number instead. If the total of the top half exceeds fifty, a bonus of thirty points will be awarded. The lower half consists of:

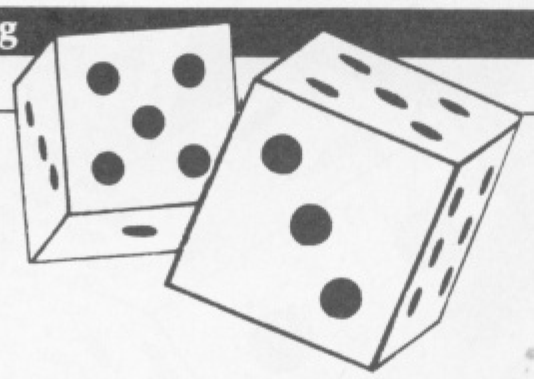
Three of a kind	: three dice with the same number — score 3 x number
Four of a kind	: four dice with the same number — score 4 x number
Low straight	: A sequence of four e.g. 1,2,3,4 — score 30
Hi Straight	: A sequence of five e.g. 1,2,3,4,5 — score 40
Full house	: Two of a number & three of another — score 25
Yehtzee	: All five dice the same — score 50 ★Jackpot★

You are given three throws before you have to 'use it'. To stop the dice from spinning press the fire button on the joystick or the space bar. After the first or second spin, you hold the individual dice by moving the marker under the required dice, by using joystick or cursor keys, and by pressing return or moving the joystick down. Press space or fire – to button start dice spinning again. If you have achieved the combination of figures before the third spin, press 'h' and you will miss out the remaining spin(s).

A cursor will now appear against the 'ones' entry which you can move to the desired place on the card by using the cursor keys or the joystick. When this is done, press the firebutton or space bar to enter, or 'h' to 'strike off' an option. Once struck off, an option cannot be used again in the same game.

When all the entries are complete, the totals will be displayed on the card. After pressing the spacebar, the master scoreboard will be shown with all the scores together. Up to six games can be played in one full match. At the end, an order will be produced displaying the winner etc. Happy throwing!

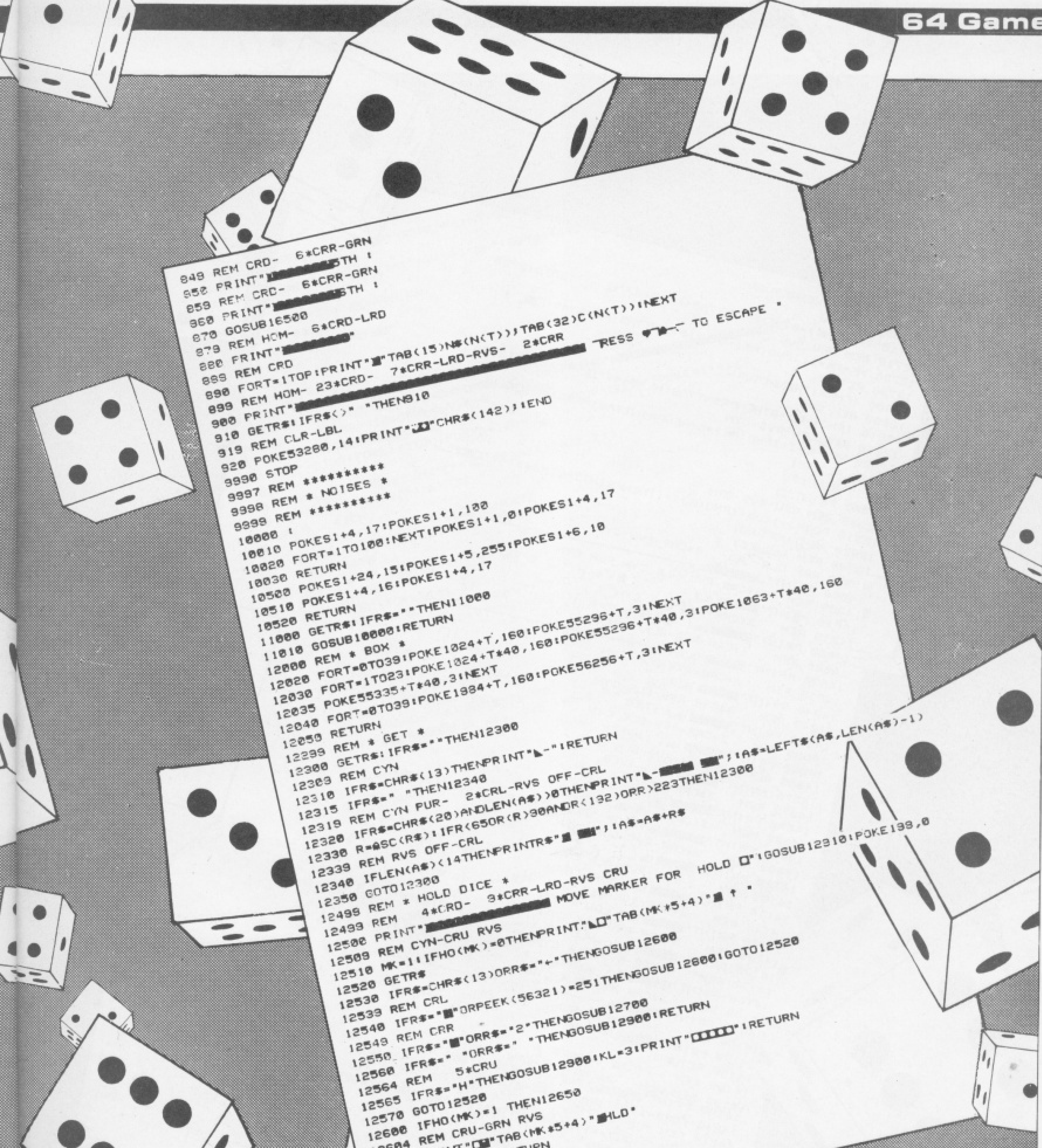
Program Listing



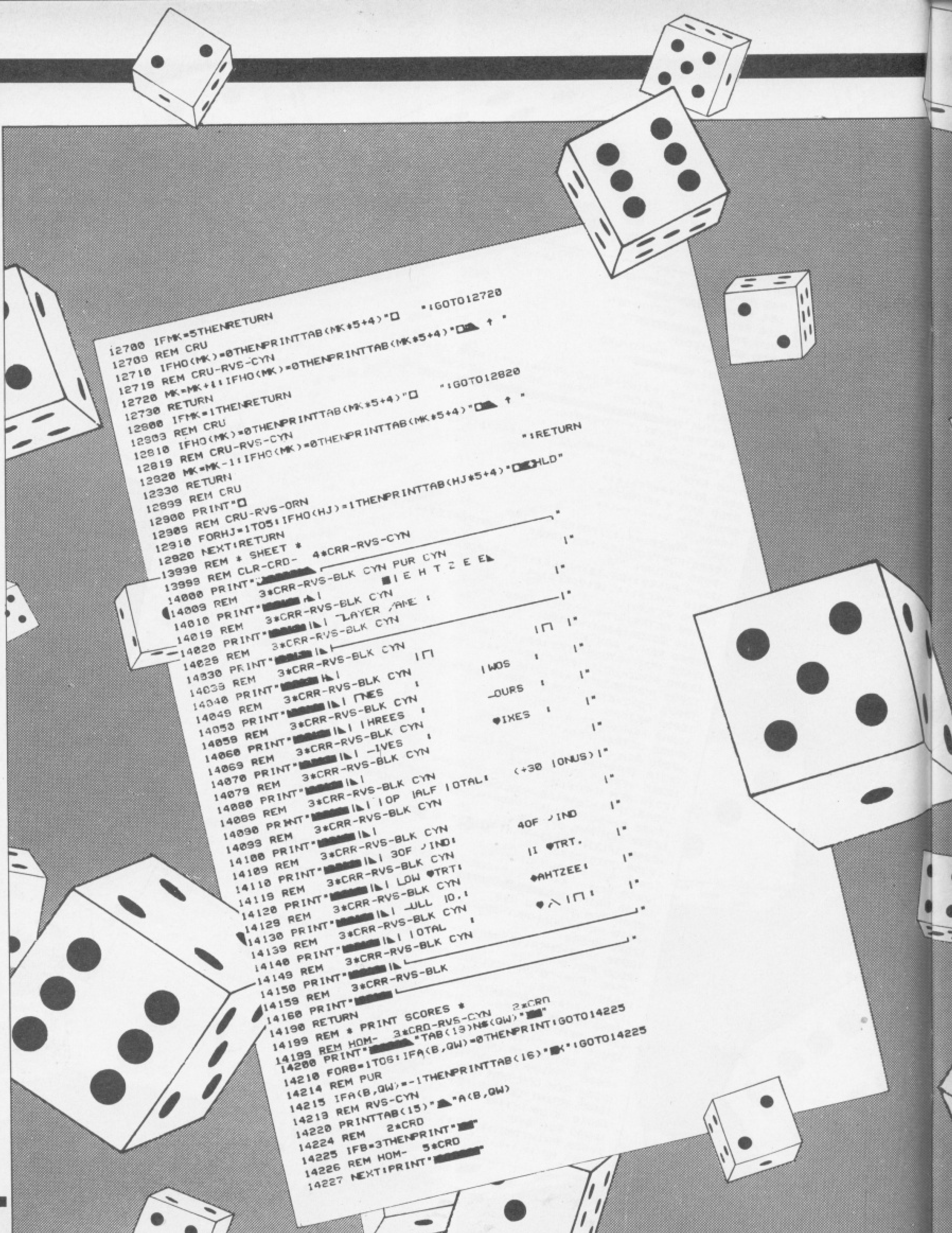
```
0 REM ********************************
1 REM *C O M P U T E R   Y E H T Z E E*
2 REM *                              *
3 REM * FOR THE COMMODORE64 COMPUTER *
4 REM *                              *
5 REM *    WRITTEN BY JAMIE CLYDE    *
6 REM *                              *
7 REM * COPYRIGHT (C)J.C DECEMBER'83 *
8 REM ********************************
9 :
10 DIMN$(5),A(12,6),D$(5,3),R%(5),A%(5),CH%(5),S(6,6),HO(5),T(4),N(4),C(4)
90 S1=54272:V=53248:GM=0:GOSUB10500
95 FORT=0TO5:FORT1=1TO3:READD$(T,T1):NEXTT1,T:FORT=0TO5:READA%(T):NEXT
99 REM CLR-TXT-CTH
100 PRINT"":POKE53280,14:POKE53281,6:POKE650,0
109 REM CRD-RVS-ORN-  3*CRR
110 PRINT"                                "
119 REM RVS-ORN-  3*CRR OFF LRD ORN RVS
120 PRINT"            -OMPUTER            "
129 REM RVS-ORN-  3*CRR OFF RVS
130 PRINT"                                "
139 REM RVS-ORN-  3*CRR YEL-CRR CRR CRR CRR CRR CRR CRR CRR CRR-ORN
140 PRINT"                                "
148 REM RVS-ORN-  3*CRR YEL-CRR CRR CRR   3*CRR CRR   2*CRR   4*CRR CRR   3*
149 REM*CRR   3*CRR-ORN
150 PRINT"                                "
159 REM RVS-ORN-  3*CRR YEL-CRR CRR   2*CRR   2*CRR   3*CRR   2*CRR   2*CRR   2*
CRR-ORN
160 PRINT"                                "
```

37

```
168 REM RVS-ORN-   3*CRR YEL-   2*CRR   2*CRR   3*CRR CRR   2*CRR   2*CRR   3*
169 REM*CRR   3*CRR   3*CRR-ORN   2*CRR CRR CRR   2*CRR   2*CRR CRR CRR
170 PRINT"                                        "
178 REM RVS-ORN-   3*CRR YEL-   2*CRR
179 REM CRR-ORN
180 PRINT"                                        "
189 REM RVS-ORN-   3*CRR OFF RVS
190 PRINT"                 WRITTEN BY \AMIE -LYDE FOR |OUR -OMMO"
199 REM RVS-ORN-   3*CRR
200 PRINT"                 - -OPYRIGHT 1983 "
201 F$="
202 F$=F$+"DORE \AGAZINE
209 REM   3*CRD-   5*CRR-CYN   4*CRU
210 PRINT"/O. OF ~LAYERS TO ~LAY?(1-6)"
211 REM CRU-   4*CRR-RVS-ORN
212 FORF=1TOLEN(F$)STEP.3:PRINT""MID$(F$,F,29)
220 GETR$:IFR$<"1"ORR$>"6"THENNEXT:GOTO212
229 REM CLR-CRD RVS-YEL CRD
230 GOSUB10000:P=VAL(R$):PRINT"";TAB(15)" /AMES:":GOSUB12000
239 REM   2*CRD-ORN CYN
240 FORG=1TOP:PRINTTAB(5)""G"   ---------------."
250 NEXT
259 REM HOM-   2*CRD   2*CRD-PUR RVS OFF-CRL
260 PRINT"":FORT=1TOP:PRINT"";TAB(10)"  ";:A$="":GOSUB12300:N$(T)=A$
270 NEXT
280 FORJ=1TOP:FORJ1=1TO12:A(J1,P)=0:NEXTJ1,J:GM=GM+1
499 REM * START THE GAME *
500 FORQW=1TOP
510 GOSUB14000:GOSUB14200:REM * BOARD *
520 GOSUB14300:REM * DICE   *
530 GOSUB14500:REM * MOVE *
540 NEXTQW:FORQ9=1TOP:FORQ8=1TO12
550 IFA(Q8,Q9)=0THENQ8=12:Q9=P:NEXTQ8,Q9:GOTO500
560 NEXTQ8,Q9
599 REM * END OF GAME *
600 O1=0:O=0:FORQW=1TOP:FORHJ=1TO3:IFA(HJ,QW)<>-1THENO=O+A(HJ,QW)
602 NEXT
603 FORHJ=7TO9:IFA(HJ,QW)<>-1THENO=O+A(HJ,QW)
610 NEXT
620 IFO>50THENO=O+30
630 FORHJ=4TO6:IFA(HJ,QW)<>-1THENO1=O1+A(HJ,QW)
632 NEXT
635 FORHJ=10TO12:IFA(HJ,QW)<>-1THENO1=O1+A(HJ,QW)
637 NEXT
640 S(GM,QW)=O+O1:GOSUB14000:GOSUB14200
649 REM HOM- 10*CRD RED-GRN
650 PRINT""TAB(21)""O
659 REM   4*CRD RED-GRN
660 PRINT""TAB(15)""O1TAB(31)S(GM,QW)
669 REM   5*CRD-   4*CRR-LRD-RVS
670 PRINT" |AME OVER-~RESS ♥PACE FOR /.~ "
680 GOSUB11000:IFR$<>" "ANDR$<>" "THEN680
690 O=0:O1=0:NEXTQW
700 GOSUB18000
710 IFR$="Y"ANDGM<>4THEN280
799 REM CLR-   4*CRD-   11*CRR-RVS-YEL
800 PRINT" |HE -INAL ~LACES: "
809 REM   3*CRD-   6*CRR-GRN
810 PRINT"1ST :
819 REM CRD-   6*CRR-GRN
820 PRINT"2ST :
829 REM CRD-   6*CRR-GRN
830 PRINT"3RD :
839 REM CRD-   5*CRR-GRN
840 PRINT"4TH :
```



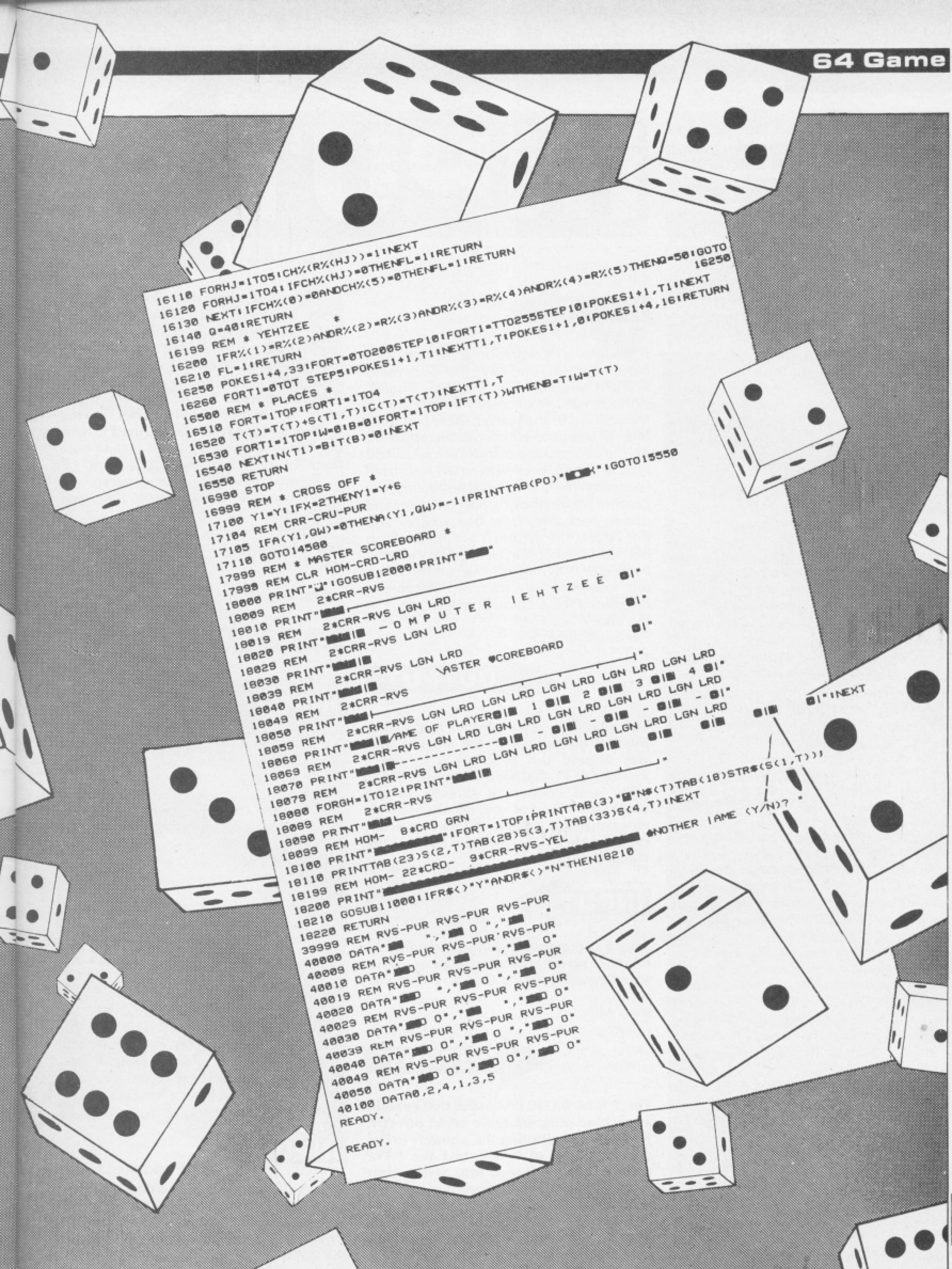
```
849 REM CRD-  6*CRR-GRN
850 PRINT"█████████5TH :
859 REM CRD-  6*CRR-GRN
860 PRINT"█████████5TH :
870 GOSUB16500
879 REM HOM-  6*CRD-LRD
880 PRINT"████████"
889 REM CRD
890 FORT=1TOP:PRINT"█"TAB(15)N$(N(T));TAB(32)C(N(T)):NEXT
899 REM HOM- 23*CRD-  7*CRR-LRD-RVS-  2*CRR
900 PRINT"████████████████████████ PRESS ██← TO ESCAPE "
910 GETR$:IFR$<>" "THEN910
919 REM CLR-LBL
920 POKE53280,14:PRINT"██"CHR$(142);:END
9990 STOP
9997 REM **********
9998 REM * NOISES *
9999 REM **********
10000 :
10010 POKES1+4,17:POKES1+1,100
10020 FORT=1TO100:NEXT:POKES1+1,0:POKES1+4,17
10030 RETURN
10500 POKES1+24,15:POKES1+5,255:POKES1+6,10
10510 POKES1+4,16:POKES1+4,17
10520 RETURN
11000 GETR$:IFR$=""THEN11000
11010 GOSUB10000:RETURN
12000 REM * BOX *
12020 FORT=0TO39:POKE1024+T,160:POKE55296+T,3:NEXT
12030 FORT=1TO23:POKE1024+T*40,160:POKE55296+T*40,3:POKE1063+T*40,160
12035 POKE55335+T*40,3:NEXT
12040 FORT=0TO39:POKE1984+T,160:POKE56256+T,3:NEXT
12050 RETURN
12299 REM * GET *
12300 GETR$:IFR$=""THEN12300
12303 REM CYN
12310 IFR$=CHR$(13)THENPRINT"▲-":RETURN
12315 IFR$=" "THEN12340
12319 REM CYN PUR-  2*CRL-RVS OFF-CRL
12320 IFR$=CHR$(20)ANDLEN(A$)>0THENPRINT"▲-████ ██";:A$=LEFT$(A$,LEN(A$)-1)
12330 R=ASC(R$):IFR<65OR(R)90ANDR<192)ORR>223THEN12300
12339 REM RVS OFF-CRL
12340 IFLEN(A$)<14THENPRINTR$"█ ██";:A$=A$+R$
12350 GOTO12300
12499 REM * HOLD DICE *
12499 REM   4*CRD-  9*CRR-LRD-RVS CRU
12500 PRINT"██████████ MOVE MARKER FOR  HOLD □":GOSUB12310:POKE198,0
12509 REM CYN-CRU RVS
12510 MK=1:IFHO(MK)=0THENPRINT"▲□"TAB(MK*5+4)"█ ↑ "
12520 GETR$
12530 IFR$=CHR$(13)ORR$="←"THENGOSUB12600
12539 REM CRL
12540 IFR$="█"ORPEEK(56321)=251THENGOSUB12800:GOTO12520
12549 REM CRR
12550 IFR$="█"ORR$="2"THENGOSUB12700
12560 IFR$=" "ORR$=" "THENGOSUB12900:RETURN
12564 REM   5*CRU
12565 IFR$="H"THENGOSUB12900:KL=3:PRINT"□□□□□":RETURN
12570 GOTO12520
12600 IFHO(MK)=1 THEN12650
12604 REM CRU-GRN RVS
12605 PRINT"□█"TAB(MK*5+4)"█HLD"
12610 HO(MK)=1:RETURN
12640 REM CRU
12650 PRINTTAB(MK*5+4)"□.    "
12660 HO(MK)=0:RETURN
```



```
12700 IFMK=5THENRETURN
12709 REM CRU
12710 IFHO(MK)=0THENPRINTTAB(MK*5+4)"□     ":GOTO12720
12719 REM CRU-RVS-CYN
12720 MK=MK+1:IFHO(MK)=0THENPRINTTAB(MK*5+4)"□■◢ ↑ "
12730 RETURN
12800 IFMK=1THENRETURN
12803 REM CRU
12810 IFHO(MK)=0THENPRINTTAB(MK*5+4)"□     ":GOTO12820
12819 REM CRU-RVS-CYN
12820 MK=MK-1:IFHO(MK)=0THENPRINTTAB(MK*5+4)"□■◢ ↑ "
12830 RETURN
12899 REM CRU
12900 PRINT"□                                   ":RETURN
12909 REM CRU-RVS-ORN
12910 FORHJ=1TO5:IFHO(HJ)=1THENPRINTTAB(HJ*5+4)"□■■HLD"
12920 NEXT:RETURN
13998 REM * SHEET *
13999 REM CLR-CRD-   4*CRR-RVS-CYN
14000 PRINT"■■■■■■■┌──────────────────────────────┐"
14009 REM    3*CRR-RVS-BLK CYN PUR CYN
14010 PRINT"■■■■■◢|              ■|E H T Z E E◣      |"
14019 REM    3*CRR-RVS-BLK CYN
14020 PRINT"■■■■■|◣| ̄LAYER /AME :─────────────────────|"
14029 REM    3*CRR-RVS-BLK CYN
14030 PRINT"■■■■■|◣|                          |┌┐  |"
14039 REM    3*CRR-RVS-BLK CYN          |┌┐
14040 PRINT"■■■■■|◣|                   |WOS   :    |"
14049 REM    3*CRR-RVS-BLK CYN  :
14050 PRINT"■■■■■|◣| ┌NES               _OURS  :    |"
14059 REM    3*CRR-RVS-BLK CYN  :
14060 PRINT"■■■■■|◣| |HREES             ♥IXES  :    |"
14069 REM    3*CRR-RVS-BLK CYN  :
14070 PRINT"■■■■■|◣| —IVES                          |"
14079 REM    3*CRR-RVS-BLK CYN
14080 PRINT"■■■■■|◣|          (+30 |ONUS)|"
14089 REM    3*CRR-RVS-BLK CYN
14090 PRINT"■■■■■|◣| |OP |ALF |OTAL:                |"
14099 REM    3*CRR-RVS-BLK CYN
14100 PRINT"■■■■■|◣|              4OF /IND         |"
14109 REM    3*CRR-RVS-BLK CYN
14110 PRINT"■■■■■|◣| 3OF /IND:    |I ♥TRT.         |"
14119 REM    3*CRR-RVS-BLK CYN
14120 PRINT"■■■■■|◣| LOW ♥TRT:    ♠AHTZEE:         |"
14129 REM    3*CRR-RVS-BLK CYN
14130 PRINT"■■■■■|◣| ┴ULL |O.:    ●/\|┌┐:          |"
14139 REM    3*CRR-RVS-BLK CYN
14140 PRINT"■■■■■|◣| |OTAL   :  ───────────────────┘"
14149 REM    3*CRR-RVS-BLK CYN
14150 PRINT"■■■■■|◣└──────────────────────────────┘"
14159 REM    3*CRR-RVS-BLK
14160 PRINT"■■■■■└──────────────────────────────"
14190 RETURN
14198 REM * PRINT SCORES *
14199 REM HOM-   3*CRD-RVS-CYN    2*CRD
14200 PRINT"■■■■◢"TAB(19)N$(QW)"■■"
14210 FORB=1TO6:IFA(B,QW)=0THENPRINT:GOTO14225
14214 REM PUR
14215 IFA(B,QW)=-1THENPRINTTAB(16)"■X":GOTO14225
14219 REM RVS-CYN
14220 PRINTTAB(15)"◢"A(B,QW)
14224 REM    2*CRD
14225 IFB=3THENPRINT"■■"
14226 REM HOM-   5*CRD
14227 NEXT:PRINT"■■■■■■"
```

```
14230 FORB=7TO12:IFA(B,QW)=0THENPRINT:GOTO14245
14234 REM PUR
14235 IFA(B,QW)=-1THENPRINTTAB(32)"■K":GOTO14245
14239 REM RVS-CYN
14240 PRINTTAB(31)"■"A(B,QW)
14244 REM   2*CRD
14245 IFB=9THENPRINT"■■"
14247 NEXT
14250 RETURN
14298 REM * DICE *
14299 REM HOM- 20*CRD
14300 FORKL=1TO5:HO(KL)=0:NEXT:FORKL=1TO3:PRINT"■■■■■■■■■■■■■■■■■■■■"
14304 REM   2*CRD- 10*CRR-RVS-GRN   4*CRU
14305 PRINT"■■■■■■■■■■ ■RESS ■■- TO ■TOP □□□□":POKES1+4,17
14307 FORZ=0TO5:FORZ1=1TO3:FORZ2=1TO5
14309 REM CRU
14310 IFHO(Z2)=0THENPRINT"□"TAB(Z2*5+4)D$(Z,Z1)
14315 POKES1+1,Z*10*Z2:NEXTZ2
14319 REM   4*CRU
14320 PRINT:NEXTZ1:PRINT"□□□□":NEXTZ
14330 GETR$:IFR$<>" "ANDR$<>" "THEN14307
14335 GOSUB10000
14340 REM * STOP DICE *
14344 REM   2*CRU
14345 PRINT"□□":POKES1+1,33
14347 FORL=1TO5:IFHO(L)=1THENNEXT:GOTO14362
14350 R%(L)=RND(1)*6:FORL1=1TO3:POKES1+1,R%(L)*10+L1:FORC=1TO10:NEXT
14359 REM   4*CRU
14360 PRINTTAB(L*5+4)D$(R%(L),L1):NEXT:PRINT"□□□□":NEXT
14362 POKES1+1,0:POKES1+4,16.
14365 IFKL<3THENGOSUB12500
14367 NEXTKL
14369 REM   4*CRD-   5*CRR RVS-ORN
14370 PRINT"■■■■■■■■■      ■■ \OVE \ARKER _OR ■CORE "
14400 RETURN
14490 STOP
14497 REM ********
14498 REM * MOVE *
14498 REM ********
14499 REM YEL CYN-RVS HOM-   5*CRD
14500 Y$="■K":Z$="■■ ":PRINT"■■■■■■■":Y=1:X=1:PO=15:POKE650,255
14510 PRINTTAB(PO)Y$
14520 GETR$:POKES1+4,17:POKES1+1,0
14529 REM CRD
14530 IFR$="■"ORR$="+"THENGOSUB15000
14539 REM CRU
14540 IFR$="□"ORPEEK(56321)=254THENGOSUB15100:GOTO14520
14543 REM CRL
14550 IFR$="■"ORPEEK(56321)=251THENGOSUB15300:GOTO14520
14553 REM CRR
14560 IFR$="■"ORR$="2"THENGOSUB15200
14570 IFR$=" "ORR$=" "THEN15500
14575 IFR$="H"THENGOTO17100
14580 POKES1+4,17:GOTO14520
15000 IFY>5THENRETURN
15004 REM CRU
15005 PRINT"□"TAB(PO)Z$
15009 REM   2*CRD
15010 IFY=3THENPRINT"■■"
15020 Y=Y+1:PRINTTAB(PO)Y$
15025 FORC=1TO255STEP10:POKES1+1,C:NEXT
15030 RETURN
15100 IFY<2THENRETURN
15104 REM CRU
15105 PRINT"□"TAB(PO)Z$
```

```
15109 REM   4*CRU
15110 IFY=4THENPRINT"□□□□"
15119 REM   2*CRU
15120 Y=Y-1:PRINT"□□"TAB(PO)Y$
15125 FORC=255TO0STEP-10:POKES1+1,C:NEXT
15130 RETURN
15200 IFX=2THENRETURN
15204 REM CRU
15205 PRINT"□"TAB(PO)Z$
15219 REM CRU
15220 X=2:PO=31:PRINTTAB(PO)"□"Y$
15225 FORC=255TO0STEP-10:POKES1+1,C:NEXT
15230 RETURN
15300 IFX=1THENRETURN
15304 REM CRU
15305 PRINT"□"TAB(PO)Z$
15319 REM CRU
15320 X=1:PO=15:PRINTTAB(PO)"□"Y$
15325 FORC=1TO255STEP10:POKES1+1,C:NEXT
15330 RETURN
15499 REM * CHECK & SCORE *
15500 FL=0:Q=0:Y1=Y:POKES1+4,16:IFX=2THENY1=6+Y
15505 IFA(Y1,QW)<>0THENPOKES1+1,10:GOTO14580
15510 IFY1<4OR(Y1>6ANDY1<10)THENGOSUB15600:GOTO15530
15520 Y2=Y1-3:IFY1>6THENY2=Y1-6
15525 ONY2GOSUB15700,15800,15900,16000,16100,16200
15530 IFFL=1THENPOKES1+1,10:GOTO14580
15539 REM CRU-RVS-CYN
15540 PRINT"□▶"TAB(PO)Q:A(Y1,QW)=Q
15549 REM HOM- 23*CRD-  8*CRR-ORN-RVS
15550 PRINT"                /EXT LAYER RESS ♥▶-.":FORT=-255TO255STEP5:POKES1+1,ABS(T):NEXT:POKES1+1,0
15560 POKES1+4,33:FORT=-255TO255STEP5:POKES1+1,ABS(T):NEXT:POKES1+1,0
15560 POKES1+4,16
15562 POKE198,0:GOSUB11000:IFR$<>" "ANDR$<>" "THEN15565
15565 RETURN
15570 STOP
15590 REM * TOP HALF  *
15599 Y2=Y1-1:IFY1>6THENY2=Y1-4
15600 FORHJ=1TO5:IFR%(HJ)=A%(Y2)THENGOSUB15650:RETURN
15610 NEXT:FL=1:RETURN.
15620 FORH8=HJTO5:IFR%(H8)=A%(Y2)THENQ=Q+A%(Y2)+1
15650 NEXT:RETURN
15660 REM * 3 OF KIND *
15699 FORHJ=5TO0STEP-1:CO=0:FORH8=1TO5
15700 IFR%(H8)=HJTHENCO=CO+1
15710 NEXT:IFCO<>3THENNEXT:FL=1:RETURN
15720 Q=3*(HJ+1):RETURN
15730 REM * LOW STRT. *
15799 FORHJ=0TO5:CH%(HJ)=0:NEXT
15800 FORHJ=1TO5:CH%(R%(HJ))=1:NEXT
15810 FORHJ=2TO3:IFCH%(HJ)=0THENFL=1:RETURN
15820 NEXT:IFCH%(1)=0ANDCH%(4)=0THENFL=1:RETURN
15830 IFCH%(0)=0ANDCH%(5)=0THENFL=1:RETURN
15835 Q=30:RETURN
15840 REM * FULL HO.  *
15899 GOSUB15700:Q=0:IFFL=1THENRETURN
15900 CO=0:FORH8=0TO5:FORH7=1TO5:IFR%(H7)=H8ANDH8<>HJTHENCO=CO+1
15910 NEXT:IFCO<>2THENNEXT:FL=1:RETURN
15920 FL=0:Q=25:RETURN
15930 REM * 4 OF KIND *
15999 FORHJ=5TO0STEP-1:CO=0:FORH8=1TO5
16000 IFR%(H8)=HJTHENCO=CO+1
16010 NEXT:IFCO<>4THENNEXT:FL=1:RETURN
16020 Q=4*(HJ+1):RETURN
16030 REM * HI STRT.  *
16099 FORHJ=0TO5:CH%(HJ)=0:NEXT
16100 FORHJ=
```



```
16110 FORHJ=1TO5:CH%(R%(HJ))=1:NEXT
16120 FORHJ=1TO4:IFCH%(HJ)=0THENFL=1:RETURN
16130 NEXT:IFCH%(0)=0ANDCH%(5)=0THENFL=1:RETURN
16140 Q=40:RETURN
16199 REM * YEHTZEE    *
16200 IFR%(1)=R%(2)ANDR%(2)=R%(3)ANDR%(3)=R%(4)ANDR%(4)=R%(5)THENQ=50:GOTO
16250
16210 FL=1:RETURN
16250 POKES1+4,33:FORT=0TO200STEP10:FORT1=TTO255STEP10:POKES1+1,T1:NEXT
16260 FORT1=0TOT STEP5:POKES1+1,T1:NEXTT1,T:POKES1+1,0:POKES1+4,16:RETURN
16500 REM * PLACES *
16510 FORT=1TOP:FORT1=1TO4
16520 T(T)=T(T)+S(T1,T):C(T)=T(T):NEXTT1,T
16530 FORT1=1TOP:W=0:B=0:FORT=1TOP:IFT(T)>WTHENB=T:W=T(T)
16540 NEXT:N(T1)=B:T(B)=0:NEXT
16550 RETURN
16990 STOP
16999 REM * CROSS OFF *
17100 Y1=Y:IFX=2THENY1=Y+6
17104 REM CRR-CRU-PUR
17105 IFA(Y1,QW)=0THENA(Y1,QW)=-1:PRINTTAB(PO)"█O█X":GOTO15550
17110 GOTO14580
17999 REM * MASTER SCOREBOARD *
17998 REM CLR HOM-CRD-LRD
18000 PRINT"♥":GOSUB12000:PRINT"███"
18009 REM   2*CRR-RVS
18010 PRINT"███┌────────────────────────────────┐
18019 REM   2*CRR-RVS LGN LRD    C O M P U T E R   I E H T Z E E  █|"
18020 PRINT"███|█  -O M P U T E R                █|"
18029 REM   2*CRR-RVS LGN LRD                  █|"
18030 PRINT"███|█
18039 REM   2*CRR-RVS LGN LRD   \ASTER ♠COREBOARD
18040 PRINT"███|█                ─┤"
18049 REM   2*CRR-RVS LGN LRD LGN LRD LGN LRD LGN LRD
18050 PRINT"███├──────────────── 1 █|█  2 █|█  3 █|█  4 █|"
18059 REM   2*CRR-RVS LGN LRD LGN LRD LGN LRD LGN LRD
18060 PRINT"███|█/AME OF PLAYER█|█  - █|█  - █|█  - █|"
18069 REM   2*CRR-RVS LGN LRD LGN LRD LGN LRD LGN LRD
18070 PRINT"███|█-------------█|█    █|█    █|█    █|█   █|":NEXT
18079 REM   2*CRR-RVS LGN LRD LGN LRD LGN LRD LGN LRD
18080 FORGH=1TO12:PRINT"███|█                   ┘"
18089 REM   2*CRR-RVS
18090 PRINT"███└──────────────────"
18099 REM HOM-  8*CRD GRN
18100 PRINT"█████████":FORT=1TOP:PRINTTAB(3)"█"N$(T)TAB(18)STR$(S(1,T));
18110 PRINTTAB(23)S(2,T)TAB(28)S(3,T)TAB(33)S(4,T):NEXT
18199 REM HOM- 22*CRD-  9*CRR-RVS-YEL
18200 PRINT"█████████████████████████████ ♠NOTHER IAME (Y/N)? "
18210 GOSUB11000:IFR$<>"Y"ANDR$<>"N"THEN18210
18220 RETURN
39999 REM RVS-PUR RVS-PUR RVS-PUR
40000 DATA"██      ","██ O ","██      "
40009 REM RVS-PUR RVS-PUR RVS-PUR
40010 DATA"██D     ","██      ","██   O"
40019 REM RVS-PUR RVS-PUR RVS-PUR
40020 DATA"██D     ","██ O ","██D O"
40029 REM RVS-PUR RVS-PUR RVS-PUR
40030 DATA"██D O","██      ","██D O"
40039 REM RVS-PUR RVS-PUR RVS-PUR
40040 DATA"██D O","██ O ","██D O"
40049 REM RVS-PUR RVS-PUR RVS-PUR
40050 DATA"██D O","██D O","██D O"
40100 DATA0,2,4,1,3,5
READY.

READY.
```

INPUT

Your Commodore computer may offer hours of fun and intellectual pursuit but it's certainly not problem free. Grahame Davies endeavours to answer some of your questions

INPUT

I have been using my VIC 20 (expanded) with a VIC 1525 printer to run self-written programs for my small business. To achieve higher memory, I have recently updated to a Commodore 64 which is advertised to run the VIC 1525 printer. Whatever I do, either in program or direct mode, I can only achieve a 'device not present' error message following printing instructions. The printer handbook does state 'User defined machine language IRQ routine should not exceed 10 milli-seconds. If this is done, the printer may give 'device not present error'. I can only think that the new 64's effectively introduce an error in this manner but I am at a loss to know how to get round it. Please can you help me.
P.D. Horne
Maidstone

OUTPUT

Two things to try here. Firstly there is a switch at the back of your 1525 right next to the serial port which selects the listen address of the printer. It is worth checking that this switch hasn't been moved when you plugged the 1525 into your 64.

Secondly because of the screen update times on the 64 the serial bus timing is slightly slower than on the VIC 20. This normally means that the bus may occasionally 'hang' when VIC 20 devices are used with a 64, but may also give device not present. To overcome this the 64 screen must be disabled while using the serial bus. This is achieved by re-setting bit 4 of register 17 in the VIC chip.

POKE 13★ 4096 +17,PEEK(13★4096+17) and (255–16)

To re-enable the screen use:

POKE 13★4096+17, PEEK(13★4096+17) or 16

Note all sprites must also be disabled. If both of these give no joy then I suggest you upgrade your printer for an MPS801 or MPS 802 which both use 64 bus timing.

INPUT

I bought a Commodore 64 on 26th November 1983 and have just discovered a slight problem on the T.V. display. Although all the necessary information is transmitted to the screen, there is a 'wavy line' appearance which is more apparent on the background colours. It looks like a type of interference pattern so I have tried re-tuning the TV set, disconnecting the cassette player, moving the computer and power pack away from one another and the TV set. The pattern is not apparent on normal TV reception but appears when a computer program is being run and when the computer is ready to receive a program. Please could you help me.
Gordon Wake
Northumberland

OUTPUT

The Interference you observe is caused by internal operations of the 64. The only way around this problem is to use a monitor or TV that has a Composite Video input and connect it directly to the video/audio port of your 64. Alternatively you could use an external RF converter, for example a video recorder by using the video-in socket.

INPUT

Does a command exist in Commodore 64 BASIC to save machine code? If not please could you tell me how to save machine code.
John Milmine
Glasgow

OUTPUT

There is no BASIC command that enables a machine program to be saved directly. However by adjusting the pointers to the start and end of BASIC text the SAVE routine can be fooled into saving your machine code. The start position is stored in locations 43 and 44. The high byte in 44 and the low byte in 43. For example, if your program starts at $5110 then the high byte is 5★16★1 or 81 and the low byte is 1★16+0 or 16. The end address is similarly stored in locations 45 and 46.

Note that when these locations have been changed you cannot access any BASIC variables so all calculations must be worked out on the screen and not stored into variables. When all four POKEs have been done the program can be saved as if it were a normal BASIC program. If you wish to re-use BASIC after saving locations 43 to 45 must be restored to their original value, so it's a good idea to note them down on a piece of paper first.

If your machine code program sits above top of BASIC i.e. $A000 then type POKE 56, 13★256: clr first.

When re-loading the program add an extra parameter 1 to the load instruction, e.g.

LOAD "M/C",1,1

This stops the machine code being re-located when it is loaded.

INPUT

A couple of weeks ago I changed my faulty VIC 20 for another one. I wrote a program on my first VIC 20 which used double sized characters using the address 36883 to drop the bottom of the screen but the screen will not drop with my second VIC 20. I also tried 36867 but without any luck. Please could you help.
Paul Hollyer
Hampshire

OUTPUT

The location 36883 is actually a reflection of the true location 36867 but should still work on any VIC 20. If bit 0 is set then the VIC chip uses character matrix of 8★16 pixels. If this does not work on your new VIC 20 then I suggest you return it to your supplier as it must have a hardware fault.

INPUT

I own a Commodore 64 and have recently bought BC BASIC which is a great improvement to the standard machine. Unfortunately BC BASIC does not support the 'paint' or 'fill' function and I cannot fill the graphic objects with the colour of my choice.

Is a routine available to make a paint or fill command? If not, how can I get round this routine in the BC BASIC or the standard 'Pokey' BASIC.
Robert Caruana
Malta

OUTPUT

A routine to perform fill is too long to publish here but I intend shortly to publish a fast algorithm for flood filling objects along with a circle algorithm both of which will be easily adapted to use the graphics commands of any BASIC extensions.

INPUT

When I move the VIC II chip to obtain extra memory, and move the start of BASIC up to location 4096, how do I locate the sprite pointers, i.e. 2040-2047?
N.Sumner
Harrogate

OUTPUT

When you move the start of BASIC the location of the sprite pointers do not change. The sprite pointers only move if you change the 16K block number that the VIC II chip accesses by changing the control lines at location $DD00 and then they move in steps of 16K along with all other video RAM locations including the screen.

INPUT

I am a fairly recent owner of the Commodore 64. The handbook has an appendix N – a Bibliography of Publications which may be of help and interest to the Commodore 64 user.

While there is no reference there to the 64 there is frequent mention of PET. How similar are these two machines?
A. Harkness
East Lothian

OUTPUT

The Commodore PET uses an identical version of Microsoft BASIC to the 64 but has no special hardware features. Reference books covering the PET will only help you master standard BASIC programs but will not help you make the most of the many features of your 64. I would therefore strongly recommend that you read some of the many good publications specifically for the 64. I would particuarly recommend the Programmers Reference Guide.

INPUT

I have written a BASIC program for sorting inputs into alpha-numerical order with a variety of facilities. One of these is the ability to LOAD files for viewing or editing. There is a minor problem with this. When searching for say, the third file on the tape (by its name). On a friend's 64, the screen lights up with each file found, a message saying what it has found appears and the screen blanks out for 5 seconds before proceeding to the next file. Although my computer searches for and loads, the correct file every time, it does not have the 5 second time delay. The screen flashes, but does not give enough time to see what is on the screen during the 'flash'.

Although it is not absolutely necessary to see what's happening during a search, it is a little worrying that, presumably, a built in function is either missing or going wrong. I returned my machine to the suppliers, but they couldn't find anything wrong and said that the 'flashing' rather than a definite time delay is normal. However, I am not convinced that two identical machines should operate differently.

I also have another problem. I hope to incorporate scrolling into my program during the delete option – i.e. to scroll the existing list in memory upward until the one to be deleted is on the screen and then break the scroll with, say, the space bar.

Page 129 of the Programmers Reference Guide says that a machine language routine is needed to shift the entire screen one character in the direction of the scroll. However, the example program on the following page does not include this machine language routine or, as far as I can see, any clue as to where to find it.

When I include the example program in my program, the first data line on the screen overprints onto the second onto the third, etc. for a few milliseconds, thus making the data difficult to look at. Is this due to the missing machine code routine or do I need to get into raster interrupt of some sort? Any help you can give me would be greatly appreciated.
R.W. Bailey
Leicester

OUTPUT

Not all 64s have exactly the same operating system; minor modifications were made with later models. For example, early 64s when clearing the screen set the foreground colour of every character to be the same as the background. But on later models the foreground colour was set to the current printing colour thus enabling characters to be POKEd to the screen without having to set the colour map first. So it is quite likely that your 64 has a later version of ROMs that do not display the filename found if it is not the name it is looking for so as to speed up the process of finding the correct file. However your 64 should display the filename for 5 seconds if it is the one requested.

When scrolling upwards there is no need for any special machine code routine for moving the screen as BASIC automatically scrolls the screen upwards when printing a carriage return or cursor down on the bottom line of the screen. The example program shown in the Programmers Reference Guide uses this feature by printing a carriage return on the bottom line of the screen in line 30 so that the screen is scrolled up ready for the message to be printed in line 50. I typed the listing in and found it to work with no modifications so it must be your implementation of the routine that is at fault. Make sure that you always position the cursor on the bottom line of the screen and print a cursor down before printing the next line. The easiest way of doing this is to define a string which consists of a home character (CHR$(9)) followed by 25 cursor downs and print this string immediately before the text you wish to print.



OUTPUT

Windows can add a professional touch to your programming. Garry Marshall shows you how to create these on the Commodore 64.

PROGRAMMING PROJECTS

RECENTLY, 'WINDOWS' HAVE BECOME popular as one of a number of ways of making computers easier to use. The mouse, of course, is another of them. Both Digital Research and MicroSoft have produced software that can support windows, and Apple's Lisa and Macintosh are also capable of maintaining them.

A window is a rectangular region of the screen in which the output from, and the results of, a particular computation can appear quite independently of the rest of the screen. With the ability to maintain and manipulate more than one window at a time, a microcomputer can appear to be carrying on more than one application simultaneously. It could, for example, show the state of a word processing program in one window, the results of the sorting operations of a database program in another, and a chart produced by a graphics program in a third.

One of the ways in which windows can make computers easier to use is by allowing information to be exchanged between two programs simply by moving it from the window of one program to that of the other. In this way, for example, moving the graph displayed in its window by a graphics program to the window of a word processing program will automatically cause it to be incorporated as an illustration in the document being produced. And a complex operation has been achieved with the greatest of ease.

The Project

This month's programming project is to write a program to make your Commodore 64 maintain a window on its screen in which text can be placed and automatically scrolled when necessary. One way of programming this is to make use of the PEEK and POKE instructions.

There are many other operations associated with windows that you can then go on to implement in order to make a windowing facility that is as flexible and useable as possible. These include being able to scroll it both downwards and upwards, panning it sideways, and moving it to another location.

When talking of scrolling and panning a window, it is usual to express any movement as that of the window over the text that is being viewed through the window. This means, for example, that panning a window to the right should give the same effect as is caused when a real window is actually moved to the right. Confusingly, perhaps, the movement of the text seen through the window will be to the left.

The solution

The first things that we must decide are the size of the window and its position on the screen. We shall place the top left corner of the window in the row with its number stored under TR and in the column with its number stored under TC. The window will be W columns wide and H rows high. This will locate the window as shown in Figure 1.

We begin the program by clearing the screen and initialising these variables with:

```
10 PRINT "♥"
20 TR=15: TC=15: W=5: H=5
```

Then we set the colour for the characters that will be displayed in the window by POKEing the same number into all the locations in the colour memory. The colour memory extends from location 55296 to 56295, and putting a number from 0 to 15 in these locations determines the colour in which any character that we POKE to the screen memory will appear. The entire colour memory is POKEd because the window may be placed anywhere on the screen. We shall use 7 to make the characters in the window yellow, and the loop for this is:

```
30 FOR K=55296 TO 56295
:POKE K,7: NEXT K
```

It is convenient to use the variable SCR to hold the address of the location in the screen memory that is mapped to the position at the top left corner of the screen. This is initialised by:

```
40 SCR=1024
```

At this stage, we can see the window by putting an inverted space (a solid yellow block) in each character position in the window. This can be done by POKEing 160 to every location in the area of screen memory that corresponds to the window. Note that the code for an inverted space is 32, the code for a space, plus 128. As the position at the top left of the screen is mapped to SCR, the position in row R and column C is mapped to the location with address SCR+40★(R–1)+C–1. Using this formula, we can highlight the window by:

```
50 FOR R=TR TO TR+H-1:
   FOR C=TC TO TC+W-1
60 AD=SCR+40*(R-1)+C-1
70 POKE AD,160
80 NEXT C: NEXT R
```

Now that we can see the window, we can think about getting input from the keyboard and placing it in the window. We shall hold the position in the window at which the next character is to be placed by storing its column under CC and its row under CR. These variables are initialised by:

```
100 CR=TR: CC=TC
```

We can get a character from the keyboard and store it under A$ by:

```
110 GET A$: IF A$=""
THEN 110
```

It can then be placed at its proper position in the window by calculating the address of the location in screen memory to which its position is mapped (as before), finding its code, and then POKEing the code into that location. The code can be found by using the function ASC; unfortunately, this function gives codes that are different from the ones used by POKE. To obtain the correct codes to use with POKE for letters, for instance, it is necessary to subtract 64 from the ASCII codes given by ASC. This gives us the lines:

```
140 AD=SCR+40*(CR-1)+CC-1
150 C=ASC(A$): IF C>64
    THEN C=C-64
160 POKE AD,C
```

We can now update the character position and return to get the next character from the keyboard with:

```
170 CC=CC+1
180 GOTO 110
```

A gap has been left between lines 110 and 140 because, as you will find if you run the program as it has developed so far, it is fine until it reaches the edge of the window; then it runs out of the window rather than wrapping round to the start of the next line in the window. We must place lines in the gap to put this right.

First, we can test for the end of the window: if the column number of the next position for a character becomes bigger than the last column of the window, then we should change the position to that at the beginning of the next row in the window. This can be done by:

```
120 IF CC>TC+W-1 THEN CC=TC:
    CR=CR+1
```

This additional line makes the program work properly until it reaches the bottom of the window. Without any further addition the new characters will extend the window, at the same width, to the bottom of the screen. To ascertain when the window is full we can find when the row of the position for the next character is bigger than the last row of the window. When this happens, we must scroll the window downwards, so that the top line vanishes and a blank line appears at the bottom. We must also change the position for the next character to the beginning of the bottom row of the window. This situation is illustrated in Figure 2.

We shall write a subroutine starting at line 1000 for the scrolling, and so the instruction we need is:

```
130 IF CR>TR+H-1 THEN
    CR=TR+H-1: GOSUB 1000
```

The scrolling can be achieved by copying the second row of the window into the first row, the third row into the second, and so on until the bottom row has been copied into the row above it. Then the bottom row must be filled with spaces. The copying is done by PEEKing each position in a row to find the code for the character there, and then POKEing this code into the location corresponding to the same column in the row above. Fortunately, PEEK and POKE use the same codes, and so there is no need to convert the codes. The subroutine is:

```
1000 FOR R=TR+1 TO TR+H-1
1010 FOR C=TC TO TC+W-1
1020 AD=SCR+40*(R-1)+C-1
1030 POKE AD-40, PEEK(AD)
1040 IF R=TR+H-1 THEN POKE
     AD,32
1050 NEXT C
1060 NEXT R
1070 RETURN
```

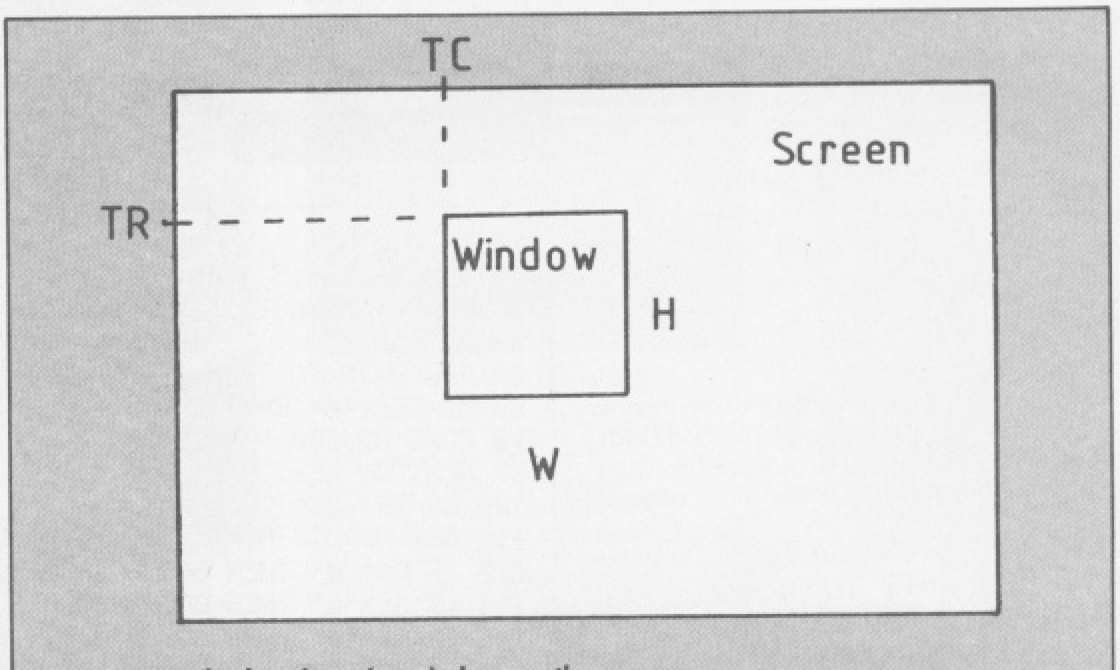


Figure 1 – The location of a window on the screen

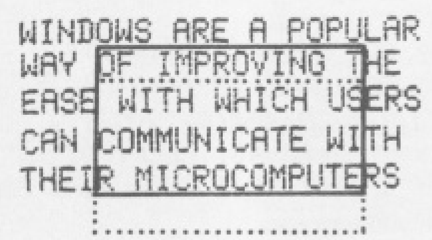


Figure 2 – Window before (bold line) and after (dotted line) scrolling downwards

We can add extra capabilities to the window program, as suggested in the problem section, but how do we call them up? One common method is to use keys that are rarely, if ever, used for other purposes; that is the approach that we shall adopt. But it must be remembered that when using the program these keys cannot be used to position the character on them in the window. The following subroutines, that start with the line numbers 2000, 3000 and 4000, respectively, implement sideways panning, moving the window and reversing the foreground and background colours of the display in a window. The characters used to invoke these capabilities are, respectively, @, # and ←.

The subroutines all work in a way that is similar to a part of the program that has already been described, so rather than repeating explanations that have already been given we shall just present the subroutines.

The additions to the program that call the subroutines that provide the extra capabilities are:

```
112 IF A$="@" THEN GOSUB
    2000: GOTO 110
114 IF A$="#" THEN GOSUB
    3000: GOTO 110
116 IF A$="←" THEN GOSUB
    4000: GOTO 110
```

The subroutine for panning the window pans it to the right, but also brings the lost left-hand column over to the right. It is:

```
2000 C=TC
2010 FOR R=TR TO TR+H-1
2020 AD=SCR+40*(R-1)+C-1
2030 B(R-TR+1)=PEEK(AD)
2040 NEXT R
2050 FOR C=TC+1 TO TC+W-1
2060 FOR R=TR TO TR+H-1
2070 AD=SCR+40*(R-1)+C-1
2080 POKE AD-1,PEEK(AD)
2090 NEXT R: NEXT C
2100 C=TC+W-1
2110 FOR R=TR TO TR+H-1
2120 AD=SCR+40*(R-1)+C-1
2130 POKE AD, B(R-TR+1)
2140 NEXT R
2150 RETURN
```

The subroutine for moving the window asks by how much the window is to be moved. Positive numbers indicate that it is to be moved to the right and downwards. Negative numbers indicate movement to the left and upwards. This subroutine assumes that the new position for the window does not overlap the old one. It is:

```
3000 INPUT "MOVE BY?(R AND
     C)"; NR,NC
3010 M=40*NR+NC
3020 FOR R=TR TO TR+H-1
3030 FOR C=TC TO TC+W-1
3040 AD=SCR+40*(R-1)+C-1
3050 POKE AD+M,PEEK(AD)
3060 NEXT C: NEXT R
3070 RETURN
```

The subroutine for inverting the window operates by adding 128 to all the codes for characters in the window that are less than 128, and subtracting 128 from the others. It is:

```
4000 FOR R=TR TO TR+H-1
4010 FOR C=TC TO TC+W-1
4020 AD=SCR+40*(R-1)+C-1
4030 IF PEEK(AD)<128 THEN
     POKE AD,PEEK(AD)+128:
     GOTO 4050
4040 POKE AD, PEEK(AD)-128
4050 NEXT C
4060 NEXT R
4070 RETURN
```

Further developments

There are many ways in which the facilities provided by the program developed here can be improved. They include the following:

★ The window itself could be highlighted at all times by showing it with a different background colour to the rest of the display.
★ The subroutine for moving a window could be amended to work properly even when the new position overlaps the old one. This would involve copying the window to a temporary store before copying it from there to the new position.
★ When a window is moved, perhaps the old one should be erased, or even replaced by what was there in the first place.
★ More than one window can be operated at the same time, and any output must be directed to the window for which it is intended. The different windows could display their contents in different colours.
★ Faster methods of scrolling and panning can be developed.
★ If a window is panned to the right and then immediately panned to the left again perhaps it should bring back the original contents rather than blank spaces.
★ When a window is moved, the new window could be made the one to which input is directed. In the present program, input always goes to the same window although copies of the current state of the window can be made all over the screen.

Program Listing

```
10 PRINT "♥"
20 TR=15: TC=15: W=5: H=5
30 FOR K=55296 TO 56295:POKE K,7: NEXT K
40 SCR=1024
50 FOR R=TR TO TR+H-1: FOR C=TC TO TC+W-1
60 AD=SCR+40*(R-1)+C-1
70 POKE AD,160
80 NEXT C: NEXT R
100 CR=TR: CC=TC
110 GET A$: IF A$="" THEN 110
112 IF A$="@" THEN GOSUB 2000: GOTO 110
114 IF A$="#" THEN GOSUB 3000: GOTO 110
116 IF A$="←" THEN GOSUB 4000: GOTO 110
120 IF CC>TC+W-1 THEN CC=TC: CR=CR+1
130 IF CR>TR+H-1 THEN CR=TR+H-1: GOSUB 1000
140 AD=SCR+40*(CR-1)+CC-1
150 C=ASC(A$): IF C>64 THEN C=C-64
160 POKE AD,C
170 CC=CC+1
180 GOTO 110
1000 FOR R=TR+1 TO TR+H-1
1010 FOR C=TC TO TC+W-1
1020 AD=SCR+40*(R-1)+C-1
1030 POKE AD-40, PEEK(AD)
1040 IF R=TR+H-1 THEN POKE AD,32
1050 NEXT C
1060 NEXT R
1070 RETURN
2000 C=TC
2010 FOR R=TR TO TR+H-1
2020 AD=SCR+40*(R-1)+C-1
2030 B(R-TR+1)=PEEK(AD)
2040 NEXT R
2050 FOR C=TC+1 TO TC+W-1
2060 FOR R=TR TO TR+H-1
2070 AD=SCR+40*(R-1)+C-1
2080 POKE AD-1,PEEK(AD)
2090 NEXT R: NEXT C
2100 C=TC+W-1
2110 FOR R=TR TO TR+H-1
2120 AD=SCR+40*(R-1)+C-1
2130 POKE AD, B(R-TR+1)
2140 NEXT R
2150 RETURN
3000 INPUT "MOVE BY?(R AND C)"; NR,NC
3010 M=40*NR+NC
3020 FOR R=TR TO TR+H-1
3030 FOR C=TC TO TC+W-1
3040 AD=SCR+40*(R-1)+C-1
3050 POKE AD+M,PEEK(AD)
3060 NEXT C: NEXT R
3070 RETURN
4000 FOR R=TR TO TR+H-1
4010 FOR C=TC TO TC+W-1
4020 AD=SCR+40*(R-1)+C-1
4030 IF PEEK(AD)<128 THEN POKE AD,PEEK(AD)+128: GOTO 4050
4040 POKE AD, PEEK(AD)-128
4050 NEXT C
4060 NEXT R
4070 RETURN
```

CONFL
CONF

Enter the field of battle and try to capture your opponent's standard in this strategic wargame for the VIC 20 from Gary Britton.



CONFLICT IS A STRATEGIC WARGAME, the object being to capture the opponent's standard. It is a two player game which requires thought and a good memory as army strengths are not revealed until an army goes into battle and mergers occur.

The game has been split into 3 parts because of the memory limitations of the VIC 20.

Program information

Line Number	Description
1-3	Initialises game and GOSUB for information on army locations.
11-33	Draws map as per program 2 with tree, house and bridge positions read from protected data statements.
35-37	Subroutine to read stored data for army locations.
40-50	POKEs armies into relevant locations and decides if player 1 or player 2 to move.
52-76	Asks for grid reference and checks the validity of the INPUT
80	Gives warning of an invalid grid reference.
84-108	Asks for a direction and checks validity of the input.
110-116	Moves army to a vacant square.
120-128	Merges two armies and removes 1 unit from the screen if no surplus remains after merger.
140-230	Sets the screen for a battle, POKEs the correct number of men in each sector and places the standard if the defending army's standard is under attack.
238-262	Loop to fight battle until one side has lost all men and takes the required action.
300	General line for picking random numbers — a subroutine used throughout the game.
400-430	If a man is to die, this subroutine activates a loop until a man is located on the screen. He is then killed off.

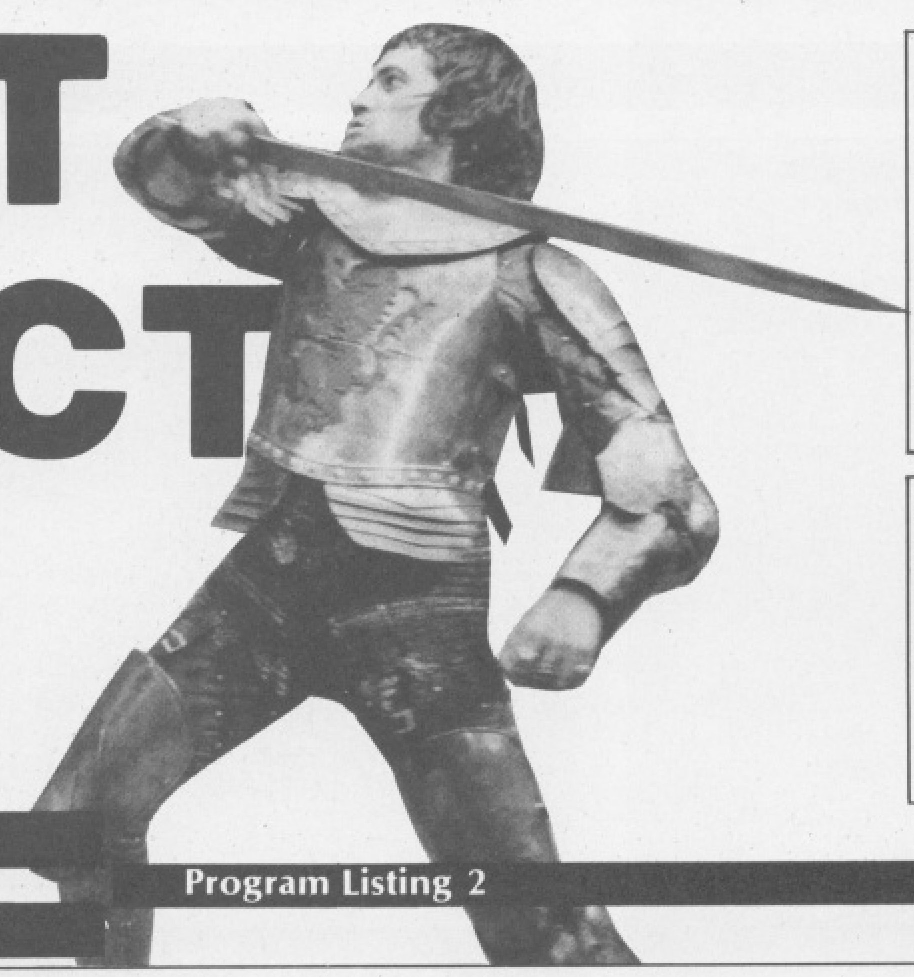
Program 1

This program contains the data statements for the user defined graphics, the introduction and instructions.

Program Listing 1

```
1 REM***PROGRAM1***
2 REM***READ DATA***
3 PRINT"■":POKE52,28:POKE56,28
4 FORI=7168TO7263:READA:POKEI,A:NEXT
8 REM***INTRODUCTION***
9 POKE36879,29:PRINT"■■■■■■■CONFLICT":PRINT"■■■■■■■--------"
10 PRINTTAB(3)"■■■BY GARY BRITTON.":PRINT"■         1984":GOSUB11:GOTO14
11 PRINT"■■■■■HIT A KEY."
12 GETA$:IFA$=""THEN12
13 PRINT"■":RETURN
14 REM***INSTRUCTIONS***
15 PRINT"  CONFLICT IS A TWO   PLAYER GAME,THE OBJECTBEING TO CAPTURE THE  ENEMY
 STANDAR
16 PRINT" WHILST DEFENDING YOUR OWN."
17 PRINT:PRINT"  EACH PLAYER BEGINS  WITH 15 ARMIES IN HIS OWN SECTOR AND THE";
18 PRINT"     STANDARD IN A FIXED   POSITION."
19 PRINT"  ON THE PLAYERS TURN,1 ARMY MAY BE MOVED BYENTERING A GRID       REFER
ENCE. IE
20 PRINT"A=COLUMN  D=ROW       AND A DIRECTION:-              N,S,E,W":GOSUB11
21 REM*****************
22 PRINT"  THE RESULT OF THIS  MOVE WILL BE 1 OF 3   ALTERNATIVES:-"
23 PRINT:PRINT"1.THE ARMY MOVES TO ANATTACK OR DEFEND       POSITION."
24 PRINT:PRINT"2.THE ARMY MEETS WITH ALLIED FORCES AND A   MERGER OCCURS."
25 PRINT:PRINT"3.THE ARMY ATTACKS THEENEMY."
26 PRINT:PRINT"A DESCRIPTION OF EACH FOLLOWS:-":GOSUB11
27 REM*****************
28 PRINT"1.SELF EXPLANATORY.":PRINT:PRINT
29 PRINT"2.ON A MERGER THE TWO ARMIES BECOME ONE WITHA MAXIMUM OF 50 UNITS,THE S
URPLUS";
30 PRINT" REMAINING AT THE OLD LOCATION."
31 PRINT:PRINT:PRINT"3.ON ATTACK THE SCREENWILL CHANGE AND SHOW  THE BATTLE TO T
HE END."
32 PRINT"THE LOSER BEING THE   ARMY WITH NO UNITS    LEFT.THE LOCATION IS";
33 PRINT"  THEN OCCUPIED BY THE  WINNER WITH HIS        REMAINING UNITS.":GOSUB11
34 REM*****************
35 POKE36879,27:POKE36869,255
36 PRINT"■■■■■■AB■ - RIVER."
37 PRINT"■■■■C■   - PLAYER 1."
38 PRINT"■■■■D■   - PLAYER 2."
39 PRINT"■■■■E■   - STANDARD."
40 PRINT"■■■■H■   - HUT."
41 PRINT"■■■■I■   - TREE."
42 PRINT"■■■■K■   - BRIDGE.":GOSUB11
43 REM*****************
44 POKE36869,240
45 PRINT"  THE NEXT SCREEN WILLSHOW THE PLAY AREA.   EACH PLAYER WILL NOW  ";
46 PRINT"LOCATE HIS ARMIES."
47 PRINT"■■■■PLEASE WAIT....."
48 LOAD
49 REM***DATA***
50 DATA0,0,0,129,194,110,60,16,0,0,32,243,94,4,0,0,8,28,114,195,129
51 DATA0,0,0,255,195,165,153,153,165,195,255,158,144,144,254,18,18
52 DATA242,0,59,63,61,32,32,32,32,112,16,16,56,63,52,56,40
53 DATA72,8,8,28,252,44,28,20,18,68,14,95,127,255,201,201,249
54 DATA28,42,73,28,42,73,8,28,66,36,32,145,66,36,44,16,255,129,255,129,255,129,2
55,0
READY.
```

ICT
LICT



Program 2

This program displays a map which is built up of the river, 5 bridges, huts and trees — all placed at random. Each player then locates his armies. The information for the location of each army, bridge, tree, etc, is stored in the unused are of memory (above the user defined graphics which is protected by lowering the top of the RAM in program 1.) This is necessary because there is no room in program 3 to fit a routine to input army locations.

Program 3

This contains the game itself (some lines contain more than 88 characters so care needs to be taken when typing it in). This program only just fits into the available memory and therefore, there is no room for REM statements. The following gives a more detailed description of the program.

Program Listing 2

```
10 REM***PROGRAM2***
11 PRINT"▯"
12 PRINT"PLEASE ENSURE THE      SHIFT LOCK KEY IS NOT DOWN."
13 PRINT"       HIT A KEY"
14 GETA$:IFA$=""THEN14
15 PRINT"▯":POKE36879,29:POKE36869,255
40 REM***DRAW MAP AND STORE***
45 REM***POSTION DATA****
50 A$="▮▮▮ABCDEFGHIJKLMNO":PRINT"▮▮▮"A$:PRINT"▮▮▮▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁]"
51 FORT=65TO79:PRINT"▮▮"CHR$(T)"| "SPC(15)" |"CHR$(T):NEXT
52 PRINT"▮▮▮▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁]":PRINTA$
56 CO=30720:S1=7800:S2=8108:L=7770:P1=1:P2=2:M=7947:S=7280
57 FORT=0TO14:R=INT(RND(1)*3):POKET+M+CO,6:POKET+M,R:NEXT
58 POKES1,5:POKES2,5:POKES1+CO,2:POKES2+CO,0
60 FORT=1TO2
62 IFT=2THENA=6:B=8:D=4:GOTO66
64 A=20:B=9:D=5
66 FORTT=1TOA
68 GOSUB76:C=Y:GOSUB76:R=Y:P=L+C+R*22
70 IFPEEK(P)<>32THEN68
72 POKEP,B:POKEP+CO,D:S=S+1:POKES,C:S=S+1:POKES,R
74 NEXTTT,T:GOTO78
76 Y=INT(RND(0)*14):RETURN
78 FORT=1TO5
79 GOSUB76:P=M+Y+22:IFPEEK(P)=32ANDPEEK(P-44)=32ANDPEEK(P-22)>0ANDPEEK(P-22)<4TH
EN81
80 GOTO79
81 POKEM+Y,11:POKEM+Y+CO,0:S=S+1:POKES,Y:NEXTT
82 PRINT"▮▮IS MAP OK (Y/N) ?":GETY$:IFY$="N"THEN15
83 IFY$<>"Y"THEN82
84 TT=1:PRINT"▮▮                 "
85 IFP3=P1THENP3=P2:DD=8:EE=15:GOTO87
86 P3=P1:DD=1:EE=7
87 PRINT"▮▮PLAYER"P3"G/R ";:GETP$:IFP$=""THEN87:REM***INPUT ARMY POSITION***
```

Program Listing 2 (cont.)

```
88 G$=G$+P$:PRINT""G$:FORT=1TO100:NEXT
89 IFLEN(G$)<2THEN87:REM***CHECK MOVE***
90 FORT=1TO2:T$=RIGHT$(G$,T):T1=ASC(LEFT$(T$,1)):T1=T1-64
91 IFT1<1ORT1>15THEN104
92 IFT=2THENT2=T1:GOTO94
93 T3=T1
94 NEXTT
95 IFT3<DDORT3>EETHEN104
96 T4=T3*22+T2+L:IFPEEK(T4)<>32THEN104
98 POKES+TT,T2:TT=TT+1:POKES+TT,T3
100 G$="":POKET4,P3+2:IFP3=1THENPOKET4+CO,2:GOTO102
101 POKET4+CO,0
102 TT=TT+1:IFTT>60THEN106
103 GOTO85
104 FORN=1TO30:POKE36879,15:POKE36876,130:PRINT" INVALID MOVE      ":NEXT:POKE3
6876,0
105 G$="":GOTO87
106 FORT=1TO2000:NEXT:PRINT"":POKE36869,240:PRINT"LOADING MAIN PROGRAM.."
107 LOAD
READY.
```

Program Listing 3

```
1 DIMB%(30,2):POKE36879,29:POKE36878,15:FORT=1TO30:B%(T,2)=30:NEXT
2 READA$:DATA"ABCDEFGHIJKLMNO"
3 A=7770:CO=30720:L1=7813:S=7280:M=7947:S1=7800:S2=8108
4 S3=S+58:FORT=0TO56STEP4:T1=1:T2=0:FORTT=1TO2:IFTT=2THENT1=16:T2=2
5 B%(T/4+T1,1)=A+(PEEK(S3+T+T2+1)*22)+PEEK(S3+T+T2):T1=0:T2=0:NEXTTT,T
11 POKE36869,255:PRINT""A$"":AD=0:V=0
13 FORT=65TO79:PRINT""CHR$(T)SPC(17)CHR$(T):NEXT:PRINTA$
15 FORT=0TO14:J=3:GOSUB300
16 POKET+M+CO,6:POKET+M,Y:NEXT
17 POKES1,5:POKES2,5:POKES1+CO,2:POKES2+CO,0
19 FORT=1TO2:H1=8:C1=4:C=1:D=11
21 IFT=2THENH1=9:C1=5:C=13:D=51
25 FORTT=CTODSTEP2:H=A+PEEK(S+TT)+(PEEK(S+TT+1)*22)
27 POKEH,H1:POKEH+CO,C1:NEXTTT:NEXTT
29 FORT=53TO57:H=M+PEEK(S+T)
31 POKEH,11:POKEH+CO,0:NEXTT
40 FORT=1TO30:T1=3:T2=2
42 IFT>15THENT1=4:T2=0
44 POKEB%(T,1),T1:POKEB%(T,1)+CO,T2:NEXT
48 P3=P3+1:IFP3=3THENP3=1:T4=0
50 IFP3=2THENT4=15
52 PRINT"PLAYER"P3"G/R   ":GETP$:IFP$=""THEN52
54 G$=G$+P$:PRINT""TAB(14)G$:FORT=1TO100:NEXT:IFLEN(G$)<2THEN52
58 P$="":FORT=1TO2:T$=RIGHT$(G$,T):T1=ASC(LEFT$(T$,1))-64:IFT1<1ORT1>15THEN80
62 IFT=2THENT2=T1:GOTO66
64 T3=T1
66 NEXTT
68 CA=T3*22+T2+A
72 G=G+1:IFG>15THENG=0:GOTO80
74 IFB%(G+T4,1)=CATHENCB=G+T4:G=0:GOTO84
76 GOTO72
80 FORN=1TO30:POKE36876,130:NEXT:POKE36876,0:G$="":GOTO52
84 PRINT"INPUT  N,S,E,W      ":GETDI$:IFDI$=""THEN84
```

Program Listing 3 (cont.)

```
86 IFDI$="N"THENDI=-22:GOTO95
88 IFDI$="S"THENDI=22:GOTO95
90 IFDI$="E"THENDI=1:GOTO95
92 IFDI$="W"THENDI=-1:GOTO95
94 GOTO84
95 G$="":D2=CA+DI:P=PEEK(D2):IFP3=1THENV1=S2:GOTO98
97 V1=S1
98 IFP3=1THENOP=4:GOTO100
99 OP=3
100 IFP=11THENP=PEEK(D2+DI):D2=D2+DI
102 IFP=OPTHEN140
103 IFD2=V1THENV=1:GOTO140
104 IFP=32THEN110
106 IFP=P3+2THEN120
108 GOTO80
110 POKECA,32:POKED2,P3+2:POKECA+CO,1:IFP3=1THENT1=2:GOTO116
114 T1=0
116 POKED2+CO,T1:B%(CB,1)=D2:GOTO48
120 F=1
122 IFD2=B%(F+T4,1)THEN126
124 F=F+1:IFF<16THEN122
126 N=B%(CB,2)+B%(F+T4,2):IFN>49THENB%(CB,2)=N-50:B%(F+T4,2)=50:GOTO48
128 POKECA,32:POKECA+CO,1:B%(CB,2)=0:B%(CB,1)=0:B%(F+T4,2)=N:GOTO48
140 PRINT"█":T1=B%(CB,2):IFV=1THENT2=10:GOTO190
145 T=1
160 T4=0
170 IFD2=B%(T+T4,1)THENT2=B%(T+T4,2):PP=T+T4:GOTO190
180 T=T+1:GOTO160
190 T=0:X=T1:MA=6:IFP3=1THENT=2
197 FORZ=1TOX
200 J=9:GOSUB300:C=Y-AD:GOSUB300:R=Y:P=L1+C*22+R:IFPEEK(P)<>32THEN200
205 POKEP,MA:POKEP+CO,T:NEXT:I=I+1
210 IFI=2THENI=0:GOTO230
220 X=T2:MA=7:AD=10:IFT=0THENT=2:GOTO225
222 T=0
225 GOTO197
230 IFV=1THENPOKEL1-5,5:POKEL1+CO-5,T
238 X=0:AD=0
240 IFT1>=XANDT2>=XTHENAA=1:BB=2:GOTO244
241 IFT1>=XTHENAA=1:BB=1
242 IFT2>=XTHENAA=2:BB=2
244 PRINT"██"TAB(3)T1TAB(13)T2:FORZ=AATOBB:J=81:GOSUB300
248 IFZ=1ANDY<=T2THENT2=T2-1:GOSUB400
250 IFZ=2ANDY<=T1THENT1=T1-1:GOSUB400
252 NEXT
254 IFT1=0THENB%(CB,1)=0:B%(CB,2)=0:B%(PP,2)=T2:GOTO11
256 IFT2=0ANDV=1THENPRINT"▀█VICTORY█":END
258 IFT2=0THENB%(CB,2)=T1:B%(CB,1)=B%(PP,1):B%(PP,1)=0:B%(PP,2)=0:GOTO11
260 AA=0:BB=0:X=X+1:IFX>T1ANDX>T2THEN238
262 GOTO240
300 Y=INT(RND(1)*J)+AD:RETURN
400 IFZ=1THENAD=10:GOTO410
405 AD=0
410 J=9:GOSUB300:C=Y-AD:GOSUB300:P=L1+22*C+Y:IFPEEK(P)=32THEN400
420 POKEP,10:POKEP+CO,2:FORT=1TO10:POKE36877,230:NEXT:POKE36877,0:POKEP,32:POKEP
+CO,1
430 PRINT"██"TAB(3)"    "TAB(13)"    ":RETURN
READY.
```

FLASH

The 1541 disc drive is slow. All is not lost. Barry Miles looks at the 1541 Flash

THERE IS CONSIDERABLE INTEREST these days in speeding up the data transfer achieved by the Commodore 1541 disc drive. A slow serial drive at the best of times, this machine can be slowed still further by reading errors caused by bad alignment. Often the bad alignment itself is caused by the type of software which makes the read/write head of the disc drive bump successively against the stop thereby knocking it out of alignment. DOS-protected software has a lot to answer for!

If you drive flashes its red light frequently when reading discs, this new device from Supersoft will not solve your problem. The first thing to do is to get your drive properly aligned by an expert. However if your disc functions satisfactorily but slowly because of its serial nature, you may well be interested in this new device. **1541 Flash** is one of a number of ways in which producers of software and hardware are seeking to aid the frustrated user who finds the delays with the 1541 disc drive absolutely interminable. Methods chosen so far by various designers include having a cartridge plugged into the cartridge slot and connected by a wire into the 64; saving your program in a special way so that it loads faster; and now a hardware system which plugs into both the 1541 and the 64 to achieve similar results. Which way you will choose to go will depend on how much faster you want the 1541 to be, and also on the depth of your pocket!

Getting started

The manual for this device warns you that it will take about half an hour to install. Whether you find this estimate correct depends on the experience which you already have in removing and replacing chips in your machine. You will need to take both the 64 and the 1541 apart, and will have to remove chips from each and mount them into special carriers. Finally, you will have to re-install these carriers in the machine. This is the sort of action which will be no hardship for people who are used to handling chips, but would be likely to frighten the newcomer to death! In any case, you should not do such work yourself if your machines are still in the warranty period: leave it to a Commodore dealer. That said, if you are experienced in handling these devices, it is not a difficult task at all to make the necessary connections.

1541 Flash is considerably more than a fast loader program. It offers a variety of additional commands including all the normal DOS or Wedge commands with which Commodore users, especially old-timers who have used the PET machines, are very familiar. In addition to this, there is a group of "Easy Everyday Commands". Perhaps the most attractive of these is the one which allows the program to be loaded just by pressing SHIFT RUN/STOP. This is further enhanced by the fact that it loads machine code programs at their correct addresses; in wother words it substitutes for the rather long-winded "LOAD"★",8,1". In order to load the program ordinarily, you simply type LOAD" and the name of the program followed by the RETURN key. Both of these commands offer real advantages to the user.

People who wish to list programs to the printer use OPEN 1:CMD1:LIST and hit RETURN.

The DOS Support or Wedge program is started by the "@" sign or the ">", whichever you find the more convenient. Initialising these commands requires you to type "SYS 655266", followed by RETURN. The switch on the cable card must be in the 1541 Flash position, (towards the computer) before DOS is started up. There are eleven standard DOS commands and three additional ones added by this program.

One of the major advantages of using this DOS Support program is that it remains in the machine after BASIC programs have been LOAD and NEWed one after another.

Thus, you can carry out Renaming, Scratching, Initialising, Newing and Copying commands with ease. In addition you can find out why the error light is on. An interesting additional command is the one which enables the DOS commands to be responded to by a device other than device 8. You simply hit "@" or "<" followed by " # " then the new device number. % followed by the program name and RETURN will load a program without re-location or changing the end links. This enables you to load machine language or graphics without disturbing your resident BASIC program.

Editing commands

This product adds 10 editing commands to the operating system of the 64. One may question whether this is a logical set of commands to include within a "speedup" module, but the commands themselves are undoubtedly convenient. With the aid of these you can move the CURSOR tothe bottom of the screen, and tab the cursor 16 spaces from the left side of the screen (which is useful with some assemblers and machine code monitors).

You can also escape from the Quote or Insert mode. You can delete a line, you can delete a screen from the line on which

the cursor is, and you can put a CHR$(27) (escape) command within strings, enabling you to send control codes to printers.

If you are to use a 1541 disc drive that does not have a 1541 flash chips installed, you can switch your 64 back to normal operation by POKE 148,64 followed by RETURN. This can also be done by throwing a manual switch on the cable card. By a somewhat longwinded command you can tell the 1541 drive to return to its normal slow method of transmission. The command is OPEN 1,8,15 "Z8 slow", followed by RETURN.

Switching the Commdodore 64 back into high speed mode is accommodated by POKE 148,0, and you may also switch the 1541 into the fast mode by OPEN 1,8,15,"Z8 fast" followed by RETURN.

Documentation

The documentation supplied is quite good and includes some interesting additional material about as previously undocumented features of the 1541, for instance invalid files with asterisks in their names can be OPENed. You can APPEND files together, other than REL files, you can SCRATCH protected files by setting bit 6 for the first byte, the file type byte in the directory entry. You can also CONCATENATE files (again I thought this was a standard command). You can also SCRATCH multiple files.

Finally, if you open a file you can use M as a file-type function like a wild card, thus OPEN 2,8,2 "filename,M,A" will open any type of file for appending.

There is a section of Advanced Programming Commands which will be of interest to machine language programmers. The 1541 can be accessed from machine language with several new commands which provide the fastest way to access the data. For instance, you can read a specified sector by single byte values and place the data in a buffer.

This command is used in place of a "U1" command. Similarly you can write data to a sector specified by single byte values for track and sector with data coming from a buffer, this is used in place of U2. You can also send data in a particular buffer to the computer at very high speed. There is a Linked List Loader whereby you provide the starting track and sector and the command will then link through and send all the following sectors fast.

In using these commands it is possible to use a special command which enables you to set up the right speed for your particular application.

A debugging aid is included in the system, accessing the Non-Maskable Interrupt. The manual gives detailed aid to the machine programmer including telling you which location will give you the information that a new 1541 ROM has been inserted.

Timings

The table shows the timings achieved with loading and saving various programs using the 1541 Flash and using exactly the same disk drive on its own. This is important because of the substantial variations in speed found when comparing two different 1541s.

Additionally the time taken to read identical sequential files using the two methods is shown. It will be seen that the acceleration is approximately three times for program loading, and twice for reading sequential files.

At the moment some games programs will not load: Flight Simulator II, and some of the Software Arts products in particular. However, Supersoft say that a new disc ROM is being developed in the U.S., which should solve this problem.

Conclusions

There is no doubt that this device fulfils a very real need in driving the 1541 at the sort of speeds to which users of the old 4040 disc drive are accustomed. It appears to do this with very considerable reliability, and offers substantial enhancement such as the DOS Support System, editing commands and other advanced programming commands into the bargain. Users whose only wish is to speed up their disc drive may feel that they are paying for rather more than this. The fact is that the hardware components make up the larger part of the cost, the throwing in of additional commands is not a significant issue.

This is an expensive device compared with others on the market which do something similar but less well. You will have to decide whether it is worth paying quite a high price to achieve the three times speed increase, or whether you will be happier with something which is cheaper, but which does not do such a good job.

For those who want the best, at the time of writing, this is it.

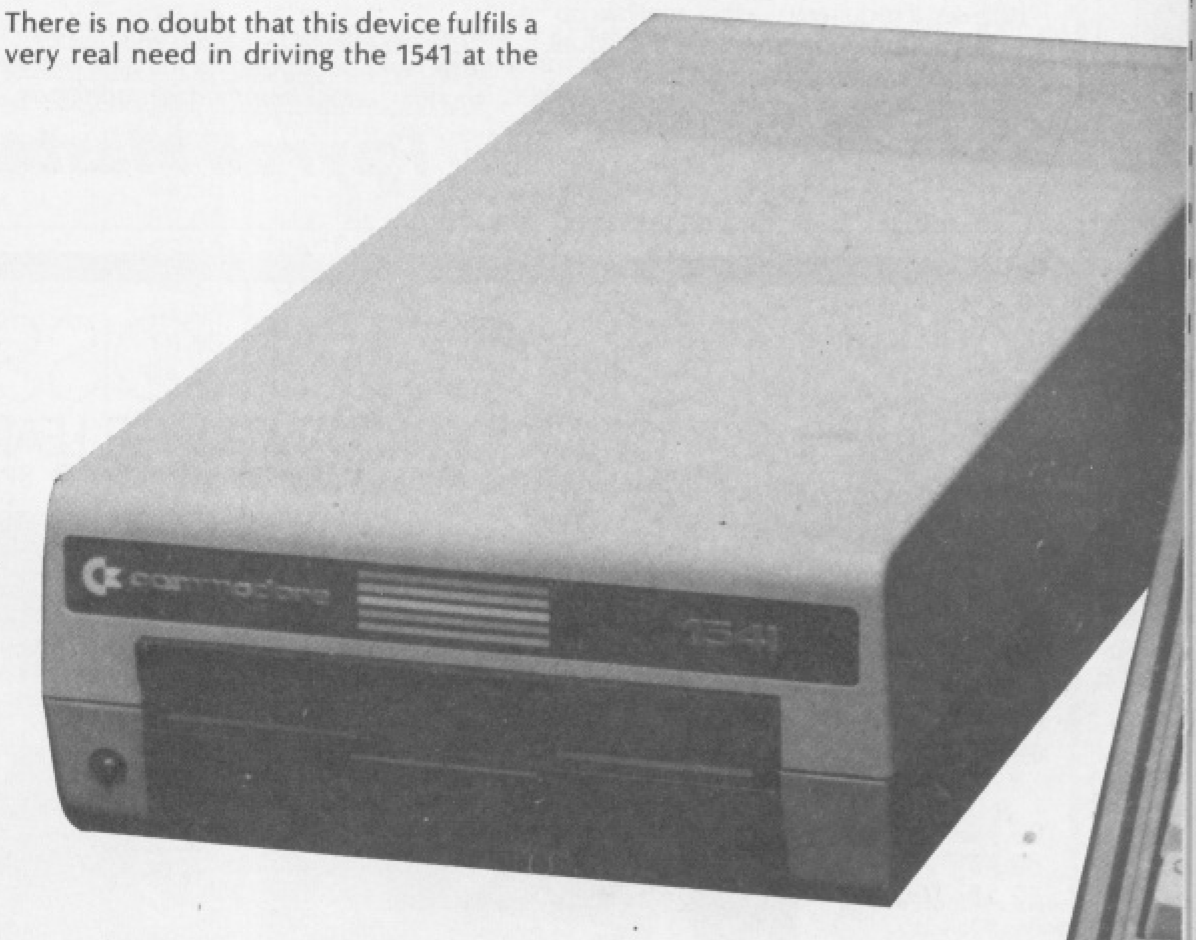


Table of Timings

Program Loads:	Standard Configuration	With 1541 Flash
Vizawrite	1min 30 secs	33 secs
Vizastar XLB	2min 15 secs	45 secs
Bristles	1min 38 secs	32 secs
Sequential File Reading: (32 blocks)	28 secs	13 secs

Payroll Master

The Master Payroll Package for the Commodore 8000 series

This outstanding payroll program is designed to Inland Revenue specifications for computerised payroll. All you require is a CBM 8032, CBM 8050 disk drive and a compatible printer (the 8296 version available soon).

The package features: all tax codes — all NI codes — automatic Statutory Sick Pay calculation — automatic holiday pay calculation (version 3 only) — 3 overtime rates of pay — 4 pre-tax adjustments (pre-set if required) — 4 after-tax adjustments (pre-set if required) — cost analysis up to 26 groups/departments — individual (payslip) and total (payday) cash analysis — system controlled numbering (Payroll No.) of employees — end of year print-out of P60's & P35's — up to 500 employees on a single disk (CBM 8050) — payslips and P60's printed on "off the shelf" continuous stationery — P35's, employees records, employees names and addresses printed on plain paper — ability to reprint the most recent payrun after the system has been closed down and the computer switched off and much more including access to "help" screens at all times — **plus exclusive 5 minute updating facility.**

Payroll Master has been extensively tested and debugged for the past twelve months, under commercial conditions.

What you get — The Payroll Master Package comes complete with floppy disk containing the suite of programs (totalling over 400K of coding) — *PLUS* — 40 page, easy to understand, comprehensive manual — *PLUS* — one box of 4,000 pre-printed two-part continuous payslips, — *PLUS* — 500 pre-printed three-part continuous P60's....

All this at an unbelievable introductory price of £399.00 + VAT (post & packaging included in price) — normally £450.00 + VAT.

This is the ideal time to change, get ready for the new tax year ... Order YOUR package *NOW!*

Delivery within 7 days of receipt — cheque with order please to:

M&M SOFTWARE
Wych Elm House, The High, Harlow,
Essex CM20 1QR.
Tel: 0279 39576

Payroll Master — *puts you in control*

FOR CONNOISSEURS

OF MODERN LANGUAGE LEARNING

COMMODORE 64 ● BBC (32K) ●ELECTRON ● SPECTRUM (48K)



Dealers, contact Lightning, Proteus, Centresoft, Microdeal or Tiger.

All titles are immediately available from good computer stores or by 24-hour mail order. Price £8.95 (overseas orders add £1)

For beginners, O-level and beyond, these best-selling programs are unique and highly successful aids to language learning. Each cassette provides a comprehensive series of vocabulary lessons and a variety of self-paced learning and test modes. All accents and special characters are clearly displayed and different colours denote masculine, feminine and neuter words to reinforce gender learning.

The create command enables new lessons in vocabulary or grammar to be entered, edited as required, then saved on tape. By using this simple yet vital feature, homework lists and exam revision can be retained indefinitely and recalled on demand.

Two cassettes are available for each language, covering thousands of words; Level A provides 16 lessons in general vocabulary; Level B provides a further 16 lessons including adjectives, adverbs and fully conjugated verb lists.

KOSMOS S·O·F·T·W·A·R·E

KOSMOS SOFTWARE LTD 1 Pilgrims Close, Harlington, DUNSTABLE, Beds. LU5 6LX Tel: (05255) 3942

The French Mistress	Level A @ £8.95 ☐	The French Mistress	Level B @ £8.95 ☐
The German Master	Level A @ £8.95 ☐	The German Master	Level B @ £8.95 ☐
The Spanish Tutor	Level A @ £8.95 ☐	The Spanish Tutor	Level B @ £8.95 ☐

KOSMOS SOFTWARE LTD.
1 Pilgrims Close, Harlington, DUNSTABLE, Beds. LU5 6LX.

Computer type ....(Commodore 64/BBC/Electron/Spectrum)
Mr/Mrs/Miss ....
Address ....
.... YC
.... Postcode ....

THE 64 SOFTWARE CENTRE
1 Princeton Street, London WC1
01-430 0954

Business accounts

Sales Ledger (Anagram) d	75.00
Purchase Ledger (Anagram) d	75.00
Cashbook (Anagram) d	75.00
Purchase Ledger (Ramtop) t	14.50
Accounts Package (Ramtop) d	75.00
Cashbook (Microsimplex) d	150.00
Cashbook (Gemini) d	64.95
Cashbook (Gemini) t	59.95
Final Accounts (Gemini) d	64.95
Final Accounts (Gemini) t	59.95
Inventory 64 (MMS) d	29.95
Stock Control (Anagram) d	75.00
Stock Control (Gemini) d	24.95
Stock Control (Gemini) t	19.95
Payroll 64 (Studio) d	34.95

Home applications

Checkbook Manager (Adamsoft) d	14.95
Home Accounts (Gemini) d	24.95
Home Accounts (Gemini) t	19.95
Home Accounts (Fieldmaster) d,t	19.95
Diary (Handic) c	19.95

Database systems

Magpie 64 (Audiogenic) d + c	44.95
Mailpro 64 (Profile) d	79.35
Oracle 64(B.I.) d	125.00
Practifile 64 (MMS) d	44.50
Simply File 64 (Simple) d	69.00
Superbase 64 (Precision) d	87.00
Figaro 64 (Saxon) d	86.25
Database (Gemini) d	24.95
Database (Gemini) c	19.95
Mailist (Gemini) d	24.95
Mailist (Gemini) t	19.95

Spreadsheets

Busicalc (Supersoft) d,t	17.95
Busicalc III (Supersoft) d	75.00
Calcresult Adv. (Handic) d	80.00
Easy Calcresult (Handic) d	49.95
Multiplan 64 (Hesware) d	104.95
Practicalc 64 (MMS) d	39.95
Practicals 64 (MMS) t	39.95
Practicalc II (MMS) d	69.95

d = disk t = tape c = cartridge

ALL IN STOCK NOW — Callers welcome (10am-6pm inc Sats)

Mail Order: Cheque or PO — add £2 for items over £30, others free. Credit cards (Access/Visa): phone or write. Immediate despatch.

The 64 Software Centre, (ref YC)
1 Princeton Street, London WC1

Please supply the following items:

1 .... **Qty**..... @£ ....
2 .... **Qty**..... @£ ....
3 .... **Qty**..... @£ ....
4 .... **Qty**..... @£ ....

TOTAL £ ....
POSTAGE £ ....
TOTAL £ ....

Name .... **Address** ....
Visa/Access Card No ....
....
Date .... **Signature** ....

Programmers' Aids & Utilities

Jetpack Compiler (DTL) d	39.95
Jetpack Compiler (DTL) t	14.50
Koalapad Tablet c	89.95
Cadpack 64 (Adamsoft) d	35.00
BC Basic (Kuma) c	57.50
Chartpak 64 (Adamsoft) d	24.95
Forth 64 (Handic) c	34.95
Forth 64 (Audiogenic) c	29.95
Forth 64 Advanced (Hesware) c	59.95
Graphics Designer (Adamsoft) d	19.95
Graphix 64 (Supersoft) t	11.50
Master 64 (Supersoft) d	71.30
Mikro Assembler (Supersoft) c	59.80
Printlink 64 (Supersoft) t	32.20
Panorama (Talent) t	17.95
Panorama (Talent) d	19.95
Simons Basic (CBM) c	47.50
Sprite Aid (Adamsoft) t	6.95
Stat 64 (Handic) c	29.95
Superdisk Utility (Adamsoft) d	12.50
Synthy 64 (Adamsoft) d	14.95
Turtle Graphics II (Hesware) c	54.95
Ultrabasic 64 (Adamsoft) d	19.95
Ultysynth (Quicksilva) t	14.95
Victree 64 (Supersoft) c	56.35
64 Doctor (MMS) d	19.95
Music Master (Supersoft) d	19.95
Master Composer (Access) d	44.95
Batic Lighting (Oasis) t	14.95
Batic Lighting (Oasis) d	19.95

Word Processors

Heswriter (Hesware) c	39.95
Paperclip 64 (B.I.) d	98.90
Simply Write 64 (Simple) d	46.00
Simply Write 64 (Simple) t	40.25
Vizawrite 64 (Viza) d	79.95
Pagewrite (Fas) d,t	19.95

Educational & Languages

Essential Italian (Dell) d	16.95
Essential Italian (Dell) t	14.95
Besieged (Sulis) t	9.95
Micro Maths (LCL) t	24.50
Lets Count (Commodore) t	9.99
BJ Bear (Commodore) t	9.99

War simulations

Eagles (SS) d	34.95
Baltic 85 (SS) d	44.95
Carrier Force (SS) d	64.95
Battle for Normandy (SS) t	14.95

Flight simulators

Interdictor Pilot (Supersoft) d	19.95
Interdictor Pilot (Supersoft) c	17.95
Flight Simulator (IFR) d	24.50
Flight Simulator II (Sublogic) d	47.50
Flight Simulator II (Sublogic) t	47.50

Business Simulations

Commodities (Blue Chip) d	50.00
Baron (Blue Chip) d	50.00
Millionaire (Blue Chip) d	50.00

Games

A large assortment on disk, cassette and cartridge.

Bills, bills, bills: those brown envelopes just keep on coming. Here is a program by Ian Rimmer which will help you keep track of your household accounts.

BILL BOARD

THIS PROGRAM MAY NOT BE ABLE TO reduce your household expenditure but it will enable you to account for it. The program is loaded in two parts: when the first part is RUN it loads in the second part and RUNs it. (The program would most likely benefit from being compiled.)

The program is menu and option driven and is in the main self-explanatory. It has been exhaustively tested and should be robust and tolerant irrespective of entries made by the user.

This is also the ideal program to use with the Turbo 64 utility listed in the December issue of *Your Commodore:* it is designed to be continually up-dated so speed in loading would be a great boon.



Exiting the program

On exit you are again asked whether a cassette or disc is being used.

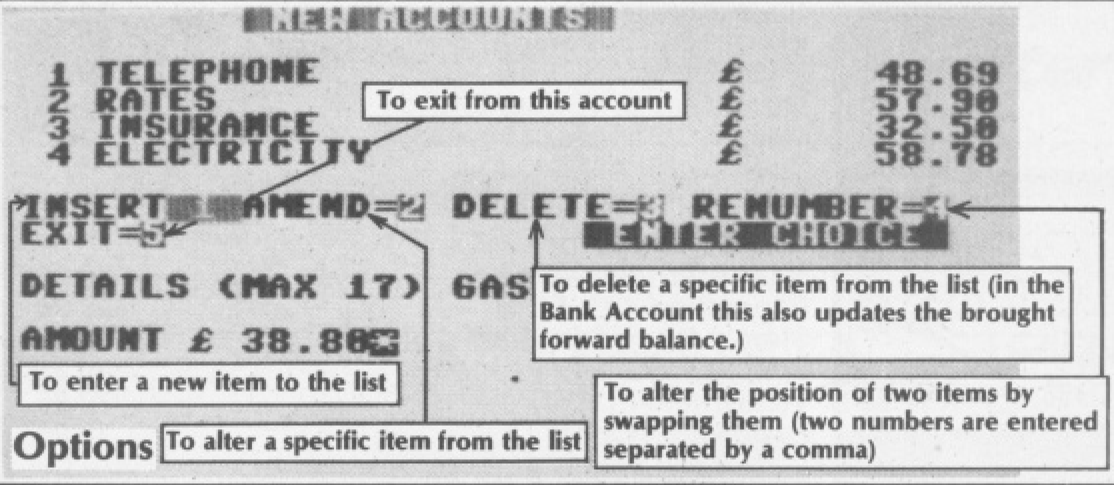
If a disc is in use you are presented with three options:

1. to delete a month from the disc.
2. to save the month in which case you are also asked about standing orders as appropriate.
3. to amend the month either to correct an error or to do another month using the details held.

If you wish to finish the program either deletes or runs itself again, depending on your choice.

Disc owners

To use on a disc drive the program should be amended as follows:
Line 220 Delete
Line 240 Amend to read "ACCOUNTS1",8



Other options

ERROR restores to the position that existed at the time an account was last entered

★ displays the next page of information. (This is only used in bills records). If it is not shown as an option it will return you to the first page if used.

CURSOR ON SCREEN to input information press Return at the end of the entry. With a flashing cursor all keys work. With a static cursor you may not enter a leading space, use the cursor keys or press return on a blank line. You can also only delete to the beginning of the field.

Take Note

- If you wish to delete an item from the bank account because you've made a mistake, you should first consider either using the error key (if possible) or the amend key. A deletion will cause the brought forward balance to change so first of all amend the item to a zero amount and then delete it.
- The delete key allows you to remove items from the bank account and credit card that appear on the statement.
- In the household accounts and standing orders you can move from one item to the next by pressing Return.
- If you make an error when calling up a function (say you press delete instead of insert), you will normally be able to cancel the error by entering a zero. You will then be given the chance to define the function required.

Program Listing 1

```
9 REM CLR-BLK
10 POKE53280,1:POKE53281,15:PRINT"█":POKE808,225
20 FORA=1236TO1356STEP40:FORB=0TO15:READC
30 POKE54272+A+B,6:POKEA+B,C:NEXTB,A
40 FORA=1428TO1548STEP40:FORB=0TO31:READC
50 POKE54272+A+B,8:POKEA+B,C:NEXTB,A
60 FORB=0TO12:READC
70 POKE55909+B,0:POKE1637+B,C:NEXTB
209 REM  18*CRD CRD
210 PRINT"██████████████████  PLEASE WAIT WHILE I LOAD":PRINT"█  THANK YOU"
220 FORC=1TO2000:NEXTC
230 POKE45,135:POKE46,86:CLR
240 LOAD"ACCOUNTS1"
250 END
260 DATA225,97,225,97,108,236,251,123,225,252,108,224,225,236,226,126
270 DATA225,252,254,97,225,97,225,97,225,236,236,224,225,252,123,96
280 DATA225,97,225,97,225,97,225,97,225,97,96,224,225,97,96,96
290 DATA124,126,124,126,96,226,226,96,124,126,96,226,124,226,226,126
300 DATA96,254,252,96,108,236,251,123,108,236,251,123,108,236,251,123
310 DATA225,97,225,97,225,252,225,97,124,251,236,126,108,236,251,123
320 DATA225,252,254,97,225,97,96,96,225,97,96,96,225,97,225,97
330 DATA225,97,225,97,225,224,224,97,96,225,97,96,124,252,98,96
340 DATA225,97,225,97,225,97,108,123,225,97,108,123,225,97,225,97
350 DATA225,97,225,97,225,97,251,97,96,225,97,96,108,123,225,97
360 DATA124,126,124,126,96,226,226,96,96,226,226,96,96,226,226,96
370 DATA96,226,226,96,124,126,124,126,96,124,126,96,96,226,226,96
380 DATA2,25,32,32,9,46,32,18,9,13,13,5,18
```

Warning

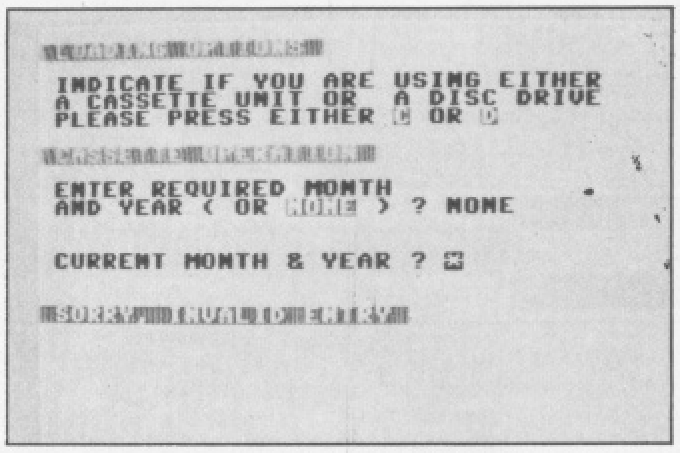
Some of the lines in the program are apparently in excess of 80 characters long; these were entered by using keyword abbreviations. Take care if the line is exactly 80 characters long for the cursor will drop to the third line (i.e. beyond the end of the logical line). Unless you cursor up to the first or second line before pressing Return the line of data you typed in will not be read into memory.

To check if this has occurred, save the programs "ACCOUNTS1" and reload it. Now check values by PEEKing locations 45 and 46 (**i.e.** –? PEEK (45); PEEK (46)); if they are not equal to 90 and 86 respectively the something has been missed.

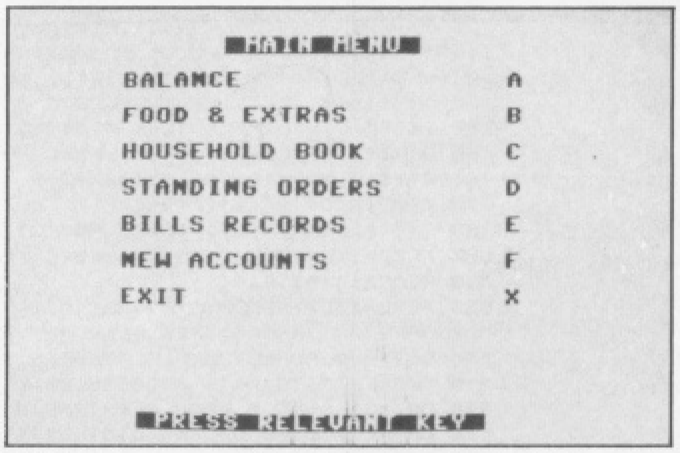
Program Listing 2

```
10 IFPEEK(53281)=255THEN50
19 REM CLR-  6*CRD RVS-RED
20 PRINT"▚██████"TAB(12)"██
24 REM RVS
25 PRINTTAB(12)"█ ILLEGAL ENTRY "
29 REM RVS OFF-BLK
30 PRINTTAB(12)"█                  ██":GOTO2440
50 IFPEEK(808)<>225THEN20
60 ME=0:R1%=1:CR=1:CO=1:CP=0:PW$=CHR$(160)
70 PRINTCHR$(8);CHR$(142)
79 REM   2*CRD-BLU BLK
80 DIMFA(13),FB(13),M$(6):M$(1)="███ PLEASE WAIT WHILE I CHECK THE INDEX█"
84 REM HOM- 13*CRD   3*CRU CRU CRU CRU
85 M$(0)="█████████████████":M$(2)="□□□                  □":M$(3)="□                  □"
89 REM CRU   2*CRU CLR CLR-CRD-GRN-RVS BLK-OFF
90 M$(4)="□                  □□":M$(5)="▚":M$(6)="▚███ FILING OPTIONS ██":FA(1)=1
100 C=12:DIMMS$(C),MF$(C),M1$(C),M2$(C),R1$(14)
110 C=20:DIMSO$(C),SO(C),HA$(C),HA(C),HB(C),ZA(C),ZA$(C)
120 C=37:DIMB1$(C),B2(C),C1$(C),C2(C),D1$(C),D2(C),E1$(C),E2(C)
130 DIMFI$(C),F2(C),G1$(C),G2(C),H1$(C),H2(C),I1$(C),I2(C),J1$(C),J2(C)
140 B4%=C:C4%=C:D4%=C:E4%=C:F4%=C:G4%=C:H4%=C:I4%=C:J4%=C
150 DIMK1$(C),K2(C),L1$(C),L2(C):K4%=C:L4%=C
160 C=73:DIMA1$(C),A2(C):A4%=C
170 DIMZ1$(C),Z2(C):Z4%=C
180 C=144:DIMM3$(C),M4$(C),M5$(C),QS%(C)
500 FORC=1TO12:READMS$(C),MF$(C):NEXT
510 READR1$(0),R1$(1)
520 FORC=1TO7:READM1$(C),M2$(C):NEXT
550 DATAJAN,JANUARY,FEB,FEBRUARY,MAR,MARCH,APR,APRIL,MAY,MAY,JUN,JUNE,JUL,JULY
560 DATAAUG,AUGUST,SEP,SEPTEMBER,OCT,OCTOBER,NOV,NOVEMBER,DEC,DECEMBER
570 DATAEXIT TO MAIN PROGRAM,BANK ACCOUNT
600 DATABALANCE,A,FOOD & EXTRAS,B,HOUSEHOLD BOOK,C
610 DATASTANDING ORDERS,D,BILLS RECORDS,E,NEW ACCOUNTS,F,EXIT,X
650 GOTO990
799 REM CRD
800 PRINT"█ INDICATE IF YOU ARE USING EITHER
809 REM RVS-RED OFF-BLK RVS-RED BLK
810 PRINT" A CASSETTE UNIT OR  A DISC DRIVE":PRINT" PLEASE PRESS EITHER ██C██ OR
 ██D█
820 GETDN$:IFDN$="C"ORDN$="D"THEN940
830 GOTO820
939 REM CRD-RVS-GRN
940 IFDN$="D"THENPRINT"███ DISC ";:GOTO960
949 REM CRD-RVS-GRN
950 PRINT"███ CASSETTE ";
959 REM BLK
960 PRINT"OPERATION █":RETURN
989 REM CLR-CRD-RVS-GRN OFF-BLK
990 POKE53280,14:PRINT"▚███ LOADING OPTIONS ██":GOSUB800
994 REM RED BLK
995 PRINT:IFCR=0THENPRINT" ENTER █*█ FOR NEXT PAGE OR"
996 PRINT" ENTER REQUIRED MONTH"
996 REM RVS-RED OFF-BLK
997 PRINT" AND YEAR ( OR ██NONE██ ) ? ";:CM$="":ZM=14:GOSUB6760:AE$=CM$
999 IFAE$="NONE"THEN1200
1000 IFAE$="0"THEN990
1003 IFAE$="*"THEN1005
1004 GOTO1009
1005 IFCR=0THENCO=CO+24
1006 IFME=0THENME=2:PRINTM$(1)
1007 GOSUB1750:POKE198,0:GOTO995
1009 MA$=AE$:GOSUB2970
1009 REM   2*CRD-RVS-RED BLK-OFF
1010 IFEC=1THENPRINT"████ SORRY, INVALID"EC$"██"TAB(24)M$(2):EC=0:GOTO997
1011 A$=LEFT$(AE$,3)+RIGHT$(AE$,2):AE$=MF$(C)+" "+RIGHT$(AE$,2):BB$="A"+A$
1013 IFDN$="C"THENOPEN2,1,0,BB$:GOTO1100
1015 OPEN15,8,15
1020 OPEN2,8,2,"0:"+BB$+",S,R"
1030 GOSUB2900:IFVAL(AF$)<20THEN1100
1035 IFVAL(AF$)=74ORVAL(AF$)=21THEN990
1050 GOTO1006
1100 INPUT#2,SO%,HA%,A3%,B3%,C3%,D3%,E3%,F3%,G3%,H3%,I3%,J3%,K3%,L3%
```

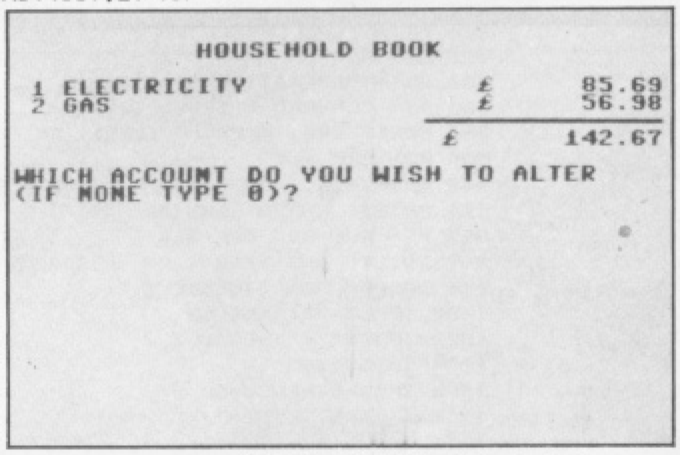
```
1105 INPUT#2,BB,SO(0),HA(0),R1%,PW$
1110 FORC=0TO12:INPUT#2,FA(C),FB(C):NEXT
1115 IFSO%>0THENFORC=1TOSO%:INPUT#2,SO$(C),SO(C):NEXT
1120 IFHA%>0THENFORC=1TOHA%:INPUT#2,HA$(C),HA(C),HB(C):NEXT
1125 IFA3%>0THENFORC=0TOA3%:INPUT#2,A1$(C),A2(C):NEXT
1130 IFB3%>0THENFORC=0TOB3%:INPUT#2,B1$(C),B2(C):NEXT
1135 IFC3%>0THENFORC=0TOC3%:INPUT#2,C1$(C),C2(C):NEXT
1140 IFD3%>0THENFORC=0TOD3%:INPUT#2,D1$(C),D2(C):NEXT
1145 IFE3%>0THENFORC=0TOE3%:INPUT#2,E1$(C),E2(C):NEXT
1150 IFF3%>0THENFORC=0TOF3%:INPUT#2,F1$(C),F2(C):NEXT
1155 IFG3%>0THENFORC=0TOG3%:INPUT#2,G1$(C),G2(C):NEXT
1160 IFH3%>0THENFORC=0TOH3%:INPUT#2,H1$(C),H2(C):NEXT
1165 IFI3%>0THENFORC=0TOI3%:INPUT#2,I1$(C),I2(C):NEXT
1170 IFJ3%>0THENFORC=0TOJ3%:INPUT#2,J1$(C),J2(C):NEXT
1175 IFK3%>0THENFORC=0TOK3%:INPUT#2,K1$(C),K2(C):NEXT
1180 IFL3%>0THENFORC=0TOL3%:INPUT#2,L1$(C),L2(C):NEXT
1185 IFR1%>1THENFORC=2TOR1%:INPUT#2,R1$(C):NEXT
1195 CLOSE2:CLOSE15
1199 REM CRD
1200 PRINT"■":IFPEEK(808)<>225THEN20
1201 PRINT" CURRENT MONTH & YEAR ? ";:CM$="":ZM=14:GOSUB6760:MA$=CM$:GOSUB2970
1201 REM   2*CRD-RVS-RED BLK-OFF
1202 IFEC=1THENPRINT"■■■■ SORRY, INVALID"EC$"■■"TAB(24)M$(2):EC=0:GOTO1201
1204 BE$=MF$(C)+" "+RIGHT$(CM$,2)
1206 IFME=0ANDDN$="D"THENPRINTM$(1):GOSUB1750:POKE198,0
1207 B$=LEFT$(BE$,3)+RIGHT$(BE$,2):DN$=""
1207 REM CLR-  2*CRD CRD
1208 IFAE$="NONE"THENA$=B$:POKE53280,3:PRINT"■■■ PLEASE SET UP ACCOUNTS REQUIRED
■":GOTO3500
1210 GOSUB1250:GOTO3000
1250 PRINT:IFCV=1ORPW$=CHR$(160)THEN1300
1254 REM CLR-CRD
1255 C=0:PRINT"■■"
1260 PRINT" ENTER PASSWORD ";:CM$="":ZM=14:GOSUB6760
1270 IFCM$=PW$THEN1300
1280 IFC=2THEN20
1290 C=C+1:PRINTTAB(16)M$(3):GOTO1260
1299 REM CRD BWN BLK
1300 IFCV=1THENPRINT"■ CURRENT PASSWORD IS ■"PW$"■"
1309 REM CRD
1310 PRINT"■ DO YOU WANT TO ALTER THE PASSWORD (Y/N)
1320 GETAN$:IFAN$="N"THEN1350
1330 IFAN$<>"Y"THEN1320
1339 REM CRD
1340 PRINT"■ NEW PASSWORD ";:CM$="":ZM=14:GOSUB6760:PW$=CM$
1350 CV=1:RETURN
1750 IFDN$="C"THENCR=1:GOTO2800
1752 CB=0:IFME=1THEN1800
1755 OPEN15,8,15:OPEN1,8,0,"$0":CJ=1:GOSUB2900:CJ=0:IFVAL(AF$))=20THEN1755
1760 FORCA=1TO33:GET#1,U$:NEXT
1765 GET#1,U$,U$
1770 GOSUB1900:IFFL=1THEN1790
1780 CB=CB+1:M3$(CB)=UB$:M4$(CB)=UC$:GOTO1770
1790 CLOSE1:CLOSE15
1800 IFME=1THEN2760
1810 CE=1:FORCC=1TO100:IFM3$(CC)=""THEN1860
1820 FORCD=1TO12
1830 MR=0:IFLEFT$(M3$(CC),4)="A"+MS$(CD)THENM5$(CE)=MF$(CD)+RIGHT$(M3$(CC),2):MR
=1
1835 IFMR=0THEN1850
1840 CD$=STR$(CD):IFCD<10THENCD$="0"+RIGHT$(CD$,1)
1845 M5$(CE)=M5$(CE)+RIGHT$(CD$,2):CE=CE+1:GOTO1855
1850 NEXTCD
1855 NEXTCC
1860 CE=CE-1:GOTO2600
1900 UB$="":FORCA=1TO4:GET#1,U$:NEXT
1905 GET#1,U$:IFU$=""THENFL=1:RETURN
1910 IFU$<>CHR$(34)THEN1905
1915 GET#1,U$
1920 IFU$=CHR$(34)THEN1930
1925 UB$=UB$+U$:GOTO1915
1930 GET#1,U$:IFU$=CHR$(32)THEN1930
1935 UC$=U$
1940 FORCA=1TO2:GET#1,U$:UC$=UC$+U$:NEXT
1945 GET#1,U$:IFU$<>""THEN1945
1950 RETURN
2000 POKE53280,12:PRINTM$(6):GOTO2028
2004 REM   2*CRD
2005 PRINT"■■ DO YOU WANT TO AMEND THIS MONTH (Y/N)
2010 GETAN$:IFAN$="N"THEN2400
2015 IFAN$<>"Y"THEN2010
2019 REM CLR-  3*CRD
2020 POKE53280,14:PRINT"■■■■ MONTH LOADED WAS"TAB(24)AE$
2024 REM   2*CRD
2025 PRINT"■■ MONTH LAST AMENDED WAS "BE$:A$=LEFT$(BE$,3)+RIGHT$(BE$,2):BE$="":G
OTO1200
2028 GOSUB800:IFDN$="D"THEN2300
```



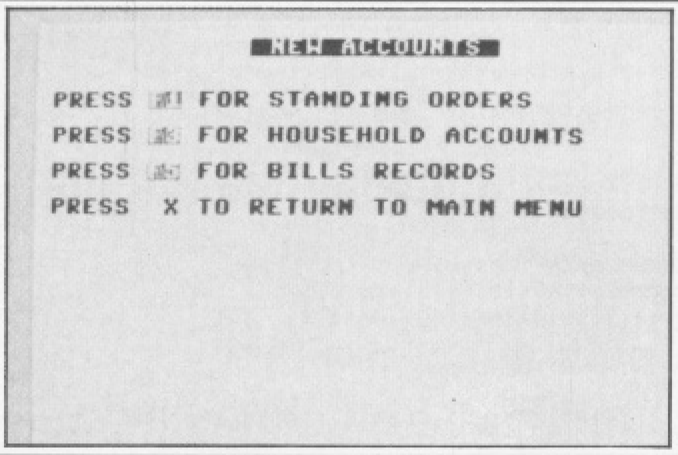
This is the first screen to greet you. You are asked whether a cassette or disc drive is being used. This affects the different ways in which a file is read or written. Also, with the disc option a record of all months on file is retained and sorted into order. Don't worry about the invalid entry – that was not the program.



This is the main menu, which shows all the different accounts and functions which the program offers. By pressing C, for example, you obtain the next screen.

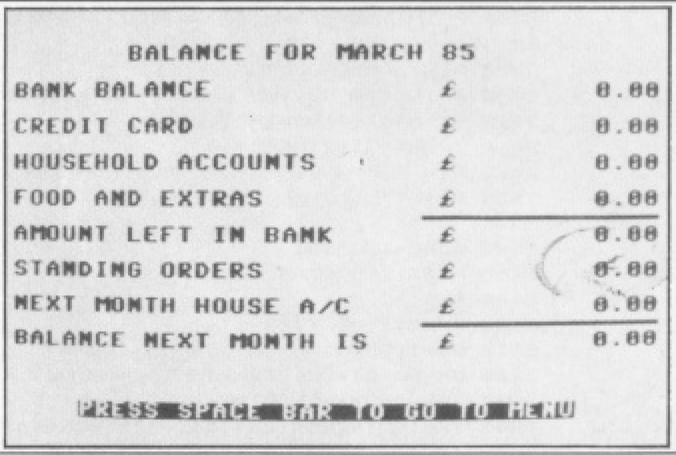


The Household Book is a typical example of one of the account options. Amounts of up £99,999.99 can be entered but if your electricity bill is that high you're going to need more help than this program can offer.
If only outgoings were really this small!



This is the menu for setting up new accounts. Household accounts (18 max.) are designed for expenses such as insurance, rates or any bill which you wish to set aside money. Standing orders (18 max.) are on a monthly basis. The bills records has one protected account for the bank account. All the other accounts can be defined (electric, telephone, credit card etc.) There are 11 user accounts.

If only outgoings were really this small!




```
2029 REM   2*CRD
2030 PRINT"■■ DO YOU WANT TO SAVE THIS MONTH (Y/N)
2035 GETAN$:IFAN$="N"THEN2005
2038 IFAN$="0"THENPRINTM$(6):GOTO2028
2040 IFAN$<>"Y"THEN2035
2043 IFB$=A$ORSO%=0THEN2065
2044 REM CRD
2045 PRINT"■ ARE THE STANDING ORDERS TO BE":PRINT" DEDUCTED FROM THE BANK BALANC
E (Y/N)
2046 REM CRU-RED BLK
2047 GETAN$:IFAN$="N"THENPRINTTAB(33)"■■ NO■":GOTO2065
2048 IFAN$<>"Y"THEN2047
2048 REM CRU-GRN
2049 PRINTTAB(33)"■■YES"
2050 GOSUB7200:FORCT=1TOSO%:Z3%=Z3%+1:IFZ3%=Z4%THENGOSUB12000
2055 Z1$(Z3%)=MID$(B$,1,5)+"   "+LEFT$(SO$(CT),17):Z2(Z3%)=-SO(CT):NEXT
2060 PM=1:GOSUB9170:GOSUB7250:FB(1)=Z6:PM=0
2064 REM CRD-GRN-RVS OFF-BLK
2065 PRINT"■■■ SAVING MONTH ■■    ";BE$:BB$="A"+B$
2068 IFDN$="C"THENOPEN2,1,1,BB$:GOTO2100
2070 OPEN15,8,15
2080 OPEN2,8,2,"@0:A"+B$+",S,W"
2090 GOSUB2900:IFVAL(AF$)>=20THENPRINTM$(6):GOTO2030
2100 CR$=CHR$(13)
2110 PRINT#2,SO%;CR$;HA%;CR$;A3%;CR$;B3%;CR$;C3%;CR$;D3%;CR$;E3%;CR$;
2120 PRINT#2,F3%;CR$;G3%;CR$;H3%;CR$;I3%;CR$;J3%;CR$;K3%;CR$;L3%;CR$;
2125 PRINT#2,BB;CR$;SO(0);CR$;HA(0);CR$;R1%;CR$;PW$;CR$
2128 FORC=0TO12:PRINT#2,FA(C);CR$;FB(C);CR$;:NEXT
2130 IFSO%>0THENFORC=1TOSO%:PRINT#2,SO$(C);CR$;SO(C);CR$;:NEXT
2140 IFHA%>0THENFORC=1TOHA%:PRINT#2,HA$(C);CR$;HA(C);CR$;HB(C);CR$;:NEXT
2150 IFA3%>0THENFORC=0TOA3%:PRINT#2,A1$(C);CR$;A2(C);CR$;:NEXT
2160 IFB3%>0THENFORC=0TOB3%:PRINT#2,B1$(C);CR$;B2(C);CR$;:NEXT
2170 IFC3%>0THENFORC=0TOC3%:PRINT#2,C1$(C);CR$;C2(C);CR$;:NEXT
2180 IFD3%>0THENFORC=0TOD3%:PRINT#2,D1$(C);CR$;D2(C);CR$;:NEXT
2190 IFE3%>0THENFORC=0TOE3%:PRINT#2,E1$(C);CR$;E2(C);CR$;:NEXT
2200 IFF3%>0THENFORC=0TOF3%:PRINT#2,F1$(C);CR$;F2(C);CR$;:NEXT
2210 IFG3%>0THENFORC=0TOG3%:PRINT#2,G1$(C);CR$;G2(C);CR$;:NEXT
2220 IFH3%>0THENFORC=0TOH3%:PRINT#2,H1$(C);CR$;H2(C);CR$;:NEXT
2230 IFI3%>0THENFORC=0TOI3%:PRINT#2,I1$(C);CR$;I2(C);CR$;:NEXT
2240 IFJ3%>0THENFORC=0TOJ3%:PRINT#2,J1$(C);CR$;J2(C);CR$;:NEXT
2250 IFK3%>0THENFORC=0TOK3%:PRINT#2,K1$(C);CR$;K2(C);CR$;:NEXT
2260 IFL3%>0THENFORC=0TOL3%:PRINT#2,L1$(C);CR$;L2(C);CR$;:NEXT
2270 IFR1%>1THENFORC=2TOR1%:PRINT#2,R1$(C);CR$;:NEXT
2280 CLOSE2:CLOSE15
2285 IFB$<>A$THENCE=CE+1:M5$(CE)=BE$+"   ":A$=B$
2290 POKE198,0:GOTO2005
2299 REM CRD
```

```
2300 CR=1:CM=0:PRINT"█ DO YOU WANT TO ERASE A MONTH (Y/N)
2305 GETAN$:IFAN$="N"THEN2030
2310 IFAN$<>"Y"THEN2305
2315 IFME=0THENME=2:PRINTM$(1)
2320 GOSUB1750:POKE198,0
2324 REM CRD-BLK RED BLK
2325 PRINT"██ ITEM NUMBER PLEASE";:IFCR=0THENPRINT",█*█ FOR NEXT PAGE";
2330 PRINT
2334 REM RVS-RED OFF-BLK
2335 INPUT" OR ██C██ TO CANCEL ";CE$:IFCE$="C"ORCE$="0"THENPRINTM$(6):GOTO2030
2340 IFCE$="*"ANDCR=0THENCO=CO+24:GOTO2320
2345 IFCE$="*"ANDCR=1THEN2320
2350 CQ=ABS(VAL(CE$)):IFCQ<1ORCQ>CETHENPRINTTAB(18)M$(3):GOTO2335
2352 IFLEFT$(M5$(CQ),1)=CHR$(30)THENPRINTTAB(18)M$(3):GOTO2335
2355 CL=LEN(M5$(CQ)):C$=LEFT$(M5$(CQ),3)+MID$(M5$(CQ),CL-3,2)
2360 OPEN15,8,15,"S0:A"+C$
2370 GOSUB2900:IFVAL(AF$)>=20THENPRINTM$(6):GOTO2300
2379 REM CRD BLU BLU BLK   2*CRU
2380 PRINT"█ █"LEFT$(M5$(CQ),CL-4);" "MID$(M5$(CQ),CL-3,2);"█ IS ERASED█        □
█"
2389 REM GRN HOM
2390 PRINTTAB(17)M$(3):M5$(CQ)="█"+M5$(CQ):CE$="":M$(5)="█":CO=CK:CLOSE15:GOTO23
20
2399 REM   2*CRD CRU
2400 PRINT"██ DO YOU WANT TO FINISH (Y/N)□
2410 GETAN$:IFAN$="N"THENRUN
2420 IFAN$<>"Y"THEN2410
2429 REM CLR-  6*CRD-RED RVS RVS RVS
2430 PRINT"█████████"TAB(15)"█             ":PRINTTAB(15)"█ GOODBYE ":PRINTTAB(15)"█
     "
2440 FORC=1TO3500
2450 NEXT:SYS64738
2460 END
2600 IFME=1THEN2760
2610 QS=1:QA=1:QB=CE:QS%(0)=1:QP=1
2620 QX=QA:QY=QB:QZ$=RIGHT$(M5$(QB),4):QX$=M5$(QB)
2630 IFQX>=QYTHEN2670
2640 IFRIGHT$(M5$(QX),4)<=QZ$THENQX=QX+1:GOTO2630
2650 IFRIGHT$(M5$(QY),4)>=QZ$THENQY=QY-1:GOTO2630
2660 QD$=M5$(QY):M5$(QY)=M5$(QX):M5$(QX)=QD$:GOTO2630
2670 M5$(QB)=M5$(QX):M5$(QX)=QX$
2680 IFQX-QA<=1THEN2720
2690 QS%(QP)=QX:QS%(QP+1)=QB:QS%(QP+2)=2:QP=QP+3
2700 QB=QX-1:GOTO2620
2710 QP=QP-3:QX=QS%(QP):QB=QS%(QP+1)
2720 IFQB-QX<=1THEN2750
2730 QS%(QP)=3:QP=QP+1:QA=QX+1:GOTO2620
2740 QP=QP-1
2750 ONQS%(QP-1)GOTO2755,2710,2740
2755 IFME=0THEN2810
2759 REM CRD
2760 PRINTM$(5);" MONTH NAMES ON FILE ARE :█"
2761 REM RVS-RED BLK
2762 IFM5$(1)=""THENPRINT"██ NO MONTH FOUND█":ME=0:FL=0:GOTO2820
2763 CR=0:CM=0:CN=0:CP=CO+23:CK=CO:IFCP>=CETHENCP=CE:CR=1
2764 REM BLU BLK
2765 FORC=COTOCP:CQ$="█"+STR$(C)+"█":IFAS<>24THENCQ$=""
2769 REM HOM-CRD
2770 CL=LEN(M5$(C)):IFCM=12THENCN=20:PRINT"██"
2780 PRINTTAB(CN+1)CQ$;TAB(5+CN)LEFT$(M5$(C),CL-4);TAB(16+CN)MID$(M5$(C),CL-3,2)
2790 CM=CM+1:NEXT
2800 IFCP=CETHENCO=1
2805 IFCM>11THENPRINTM$(8)
2810 ME=1
2819 REM CLR
2820 M$(5)="█":RETURN
2900 AF$=CHR$(0)
2910 INPUT#15,AF$,AG$,AH$,AI$
2920 IFVAL(AF$)<20THEN2960
2929 REM   2*CRD-RVS-RED BLK-OFF
2930 PRINT"████ FILE ERROR ██          "
2932 PRINT" CODE"TAB(27)"TRACK  BLOCK":PRINT" "AF$,AG$,AH$"  "AI$
2935 IFVAL(AF$)=74ORVAL(AF$)=21THEN2950
2940 PRINT#15,"I0"
2949 REM CRD-RVS-RED OFF
2950 IFCJ=1THENPRINT"████ TAKE CORRECTIVE ACTION THEN █"
2952 REM CRD-BLU BLK-CRU
2953 PRINT"██ PRESS ANY KEY TO CONTINUE█□"
2955 WAIT197,64:WAIT197,64,64:POKE198,0:CLOSE1:CLOSE2:CLOSE15
2960 RETURN
2970 IFLEN(MA$)<5THENEC=1:EC$=" ENTRY ":GOTO2995
2975 FORC=1TO2:MB$=MID$(RIGHT$(MA$,2),C,1):IFMB$<"0"ORMB$>"9"THENEC=1:EC$=" YEAR
 ":GOTO2995
2980 NEXT:MC$=LEFT$(MA$,3):FORC=1TO12:IFMC$=M6$(C)THEN2995
2985 NEXT:EC=1:EC$=" MONTH "
2995 RETURN
```

```
3000 POKE53280,13:IFPEEK(808)<>225THEN20
3004 REM CLR-CRD RVS
3005 NO=0:PRINT"■■"TAB(14)"■ MAIN MENU "
3009 REM CRD
3010 FORC=1TO9:PRINT"■"TAB(8)M1$(C);TAB(30)M2$(C):NEXT
3019 REM   2*CRD-RVS OFF
3020 PRINTTAB(9)"■■■ PRESS RELEVANT KEY ■"
3030 GETAN$:IFAN$=""THEN3030
3040 AS=ASC(AN$)-64:IFAS=24GOTO2000
3050 IFAS<1ORAS>6THEN3030
3060 ONASGOTO4000,4350,4500,4750,7000,3505
3499 REM CLR-  2*CRD
3500 PRINT"■■■":GOSUB1300
3505 POKE53280,4
3509 REM CLR-CRD RVS OFF   2*CRD RED-RVS OFF-BLK
3510 PRINT"■■"TAB(13)"■ NEW ACCOUNTS ■":PRINT"■■ PRESS ■■F1■■ FOR STANDING ORDER
S
3514 REM CRD RVS-RED OFF-BLK CRD RVS-RED OFF-BLK
3515 PRINT"■ PRESS ■■F3■■ FOR HOUSEHOLD ACCOUNTS":PRINT"■ PRESS ■■F5■■ FOR BILLS
 RECORDS
3519 REM CRD
3520 PRINT"■ PRESS  X TO RETURN TO MAIN MENU
3530 GETAN$:IFAN$=""THEN3530
3535 IFAN$="X"THEN3000
3539 REM F1
3540 IFAN$="■"THEN3560
3544 REM F3
3545 IFAN$="■"THENF=3:GOTO3600
3549 REM F5
3550 IFAN$="■"THENF=5:GOSUB8500:GOTO3505
3555 GOTO3530
3559 REM CLR RVS-BLU OFF-BLK-CRD
3560 F=1:PRINT"■"TAB(10)"■■ "M1$(4)" ■■■":IFSO%=0THEN3585
3570 FORC=1TOSO%:IFC<10THENPRINT" ";
3580 PRINTCTAB(4)SO$(C)TAB(28)"£";:Z9=SO(C):GOSUB13750:NEXT
3585 N=SO%:GOSUB6000
3590 IFAN=5THENN=0:GOTO3505
3595 GOTO3560
3599 REM CLR RVS-BLU OFF-BLK-CRD
3600 PRINT"■"TAB(10)"■■ "M1$(3)" ■■■":IFHA%=0THEN3625
3610 FORC=1TOHA%:IFC<10THENPRINT" ";
3620 PRINTCTAB(4)HA$(C)TAB(28)"£";:Z9=HA(C):GOSUB13750:NEXT
3625 N=HA%:GOSUB6000
3630 IFAN=5THENN=0:GOTO3505
3635 GOTO3600
3999 REM CLR-  2*CRD
4000 PRINT"■■■"TAB(8)"BALANCE FOR "+BE$
4004 REM CRD
4005 PRINT"■ BANK BALANCE"TAB(26)"£";:FA(0)=1:GOSUB4250:A6=Z9:GOSUB13749
4007 REM CRD
4008 PRINT"■ CREDIT CARD"TAB(26)"£";:FA(0)=2:GOSUB4250:B6=Z9:Z9=-Z9:GOSUB13749
4009 REM CRD
4010 PRINT"■ HOUSEHOLD ACCOUNTS"TAB(26)"£";:Z9=-HA(0):GOSUB13749
4019 REM CRD
4020 PRINT"■ FOOD AND EXTRAS"TAB(26)"£";:Z9=-BB:GOSUB13749
4030 PRINTTAB(25)"──────────────":BC=A6-(HA(0)+B6+BB)
4040 PRINT" AMOUNT LEFT IN BANK"TAB(26)"£";:Z9=BC:GOSUB13749
4049 REM CRD
4050 PRINT"■ STANDING ORDERS"TAB(26)"£";:Z9=-SO(0):GOSUB13749
4060 HB(0)=0:FORC=1TOHA%:HB(0)=HB(0)+HB(C):NEXT
4069 REM CRD
4070 PRINT"■ NEXT MONTH HOUSE A/C"TAB(26)"£";:Z9=-HB(0):GOSUB13749
4080 PRINTTAB(25)"──────────────":BC=BC-(SO(0)+HB(0))
4090 PRINT" BALANCE NEXT MONTH IS"TAB(26)"£";:Z9=BC:GOSUB13749
4099 REM   3*CRD-BLK-RVS OFF
4100 PRINTTAB(5)"■■■■■PRESS SPACE BAR TO GO TO MENU■
4110 GETAN$:IFAN$<>" "THEN4110
4120 GOTO3000
4250 Z9=0:FORC=1TOR1%:IFFA(0)=FA(C)THENZ9=Z9+FB(C)
4260 NEXTC:RETURN
4349 REM CLR-  3*CRD
4350 PRINT"■■■■ THE FOOD & EXTRAS ARE"TAB(28)"£";:Z9=BB:GOSUB13750
4359 REM   2*CRD
4360 PRINT"■■ DO YOU WISH TO ALTER THE ACCOUNT (Y/N)
4370 GETAN$:IFAN$="N"THEN3000
4380 IFAN$<>"Y"THEN4370
4390 Z9=BB:GOSUB6500:BB=AA:GOTO4350
4499 REM CLR-CRD CRD
4500 PRINT"■■"TAB(12)M1$(3)"■":GOSUB6250:CA$=""
4539 REM CRD
4540 PRINT"■ WHICH ACCOUNT DO YOU WISH TO ALTER
4550 INPUT" (IF NONE TYPE 0)";CA$:CA=ABS(VAL(CA$)):IFCA=0THEN3000
4560 IFCA>HA%THENPRINTTAB(18)M$(3):GOTO4550
4564 REM CLR-CRD CRD
4565 PRINT"■■"TAB(12)M1$(3)"■":IFCA<10THENPRINT" ";
4570 PRINTCATAB(4)HA$(CA)TAB(28)"£";:Z9=HA(CA):GOSUB13750
```

```
4579 REM CRD
4580 PRINT"■ PLEASE ADD ON PER MONTH"TAB(28)"£";:Z9=HB(CA):GOSUB13750
4589 REM CRD RVS-RED OFF-BLK
4590 PRINT"■ TO CHANGE ADVISED AMOUNT PRESS ■C■"
4594 REM RVS-RED OFF-BLK-CRU
4595 PRINT" TO ADD IN ADVISED AMOUNT PRESS ■P■■"
4600 Z9=HA(CA):GOSUB6500:IFCF=2THENHB(CA)=AA:CF=0:GOTO4620
4610 HA(CA)=AA:CA$=STR$(CA+1)
4619 REM CLR-CRD CRD
4620 PRINT"■■"TAB(12)M1$(3)"■":GOSUB6250:GOTO4540
4749 REM CLR-CRD CRD
4750 PRINT"■■"TAB(12)M1$(4)"■":GOSUB6250:CA$=""
4789 REM CRD
4790 PRINT"■ WHICH ORDER DO YOU WISH TO ALTER
4800 CA$="":INPUT" (IF NONE TYPE 0)";CA$:CA=ABS(VAL(CA$)):IFCA=0THEN3000
4810 IFCA>SO%THENPRINTTAB(18)M$(3):GOTO4800
4819 REM CLR-CRD CRD
4820 PRINT"■■"TAB(12)M1$(4)"■":IFCA<10THENPRINT" ";
4830 PRINTCATAB(3)SO$(CA)TAB(27)"£";:Z9=SO(CA):GOSUB13750
4839 REM CRD CLR-CRD CRD
4840 AA=0:CF=2:PRINTTAB(8)"■ NEW";:GOSUB6550:SO(CA)=AA:PRINT"■■"TAB(12)M1$(4)"■"
:GOSUB6250:GOTO4790
5999 REM CRD RVS-RED BLK-OFF RVS-RED BLK-OFF RVS-RED BLK-OFF RVS-RED BLK-OFF
6000 PRINT"■ INSERT=■1■ AMEND=■2■ DELETE=■3■ RENUMBER=■4■
6004 REM RVS-RED BLK-OFF RVS-BWN BLK
6005 PRINT" EXIT=■5■"TAB(23)"■ ENTER CHOICE ■":NO$="":ZU$=""
6010 GETAN$:AN=VAL(AN$):IFAN<1ORAN>5GOTO6010
6013 IFAN=5THEN6090
6015 ONANGOSUB6093,6093,6096,6096
6020 IFAN=1THENNO=N+1:GOTO6040
6025 IFAN=4THEN6040
6029 REM CRD
6030 PRINT"■ NUMBER PLEASE ? ";:CM$="":ZM=12:GOSUB6760:NO$=CM$:NO=ABS(VAL(NO$))
6032 IFNO>NTHENPRINTTAB(17)M$(4):GOTO6030
6035 IFNO=0THEN6090
6040 ONANGOSUB6100,6100,6200,6230
6060 IFNO=19THEN6090
6070 IFF=1THENSO$(NO)=ZA$(NO):SO(NO)=ZA(NO):IFAN=1THENSO%=NO:N=SO%
6075 IFF=1THENSO(0)=0:FORC=1TOSO%:SO(0)=SO(0)+SO(C):NEXT
6080 IFF=3THENHA$(NO)=ZA$(NO):IFAN=1THENHA%=NO:N=HA%
6090 NO=0:RETURN
6092 REM   2*CRU RVS-GRN CRL BLK-OFF-CRD
6093 PRINT"■■"TAB(AN*8-1)"■"AN"■ ■■":RETURN
6095 REM   2*CRU RVS-GRN CRL BLK-OFF-CRD
6096 PRINT"■■"TAB(AN*11-9)"■"AN"■ ■■":RETURN
6099 REM RVS-BLU OFF-BLK
6100 IFNO=19THENPRINT"■ SORRY, FILES FULL ■":WAIT197,64:WAIT197,64,64:POKE198,
0:RETURN
6110 ZM=17:GOSUB6750:ZA$(NO)=CM$:PRINT
6120 IFF=3THEN6140
6130 PRINT" AMOUNT £ ";:GOSUB13545
6140 RETURN
6200 IFF=3THEN6220
6205 SO$(NO)="":SO(NO)=0
6210 IFNO=SO%+1THENSO%=SO%-1:N=SO%:RETURN
6215 SO$(NO)=SO$(NO+1):SO(NO)=SO(NO+1):NO=NO+1:GOTO6210
6220 HA$(NO)="":HA(NO)=0:HB(NO)=0
6223 IFNO=HA%+1THENHA%=HA%-1:HB%=HB%-1:N=HA%:RETURN
6226 HA$(NO)=HA$(NO+1):HA(NO)=HA(NO+1):HB(NO)=HB(NO+1):NO=NO+1:GOTO6223
6228 REM CRD RED BLK RED BLK
6230 INPUT"■ THE TWO NUMBERS ARE ( ■*■ , ■*■ ) ";ZU$,ZV$
6231 ZU=ABS(VAL(ZU$)):ZV=ABS(VAL(ZV$)):IFZU=0ORZV=0THENRETURN
6232 IFF=3GOTO6240
6233 REM RVS-RED BLK-OFF
6234 IFZU>SO%ORZV>SO%THENPRINT"■ INCORRECT NUMBER ■":GOTO6230
6236 ZW$=SO$(ZU):ZW=SO(ZU):SO$(ZU)=SO$(ZV)
6238 SO(ZU)=SO(ZV):SO$(ZV)=ZW$:SO(ZV)=ZW:RETURN
6239 REM RVS-RED OFF-BLK
6240 IFZU>HA%ORZV>HA%THENPRINT"■ INCORRECT NUMBER ■":GOTO6230
6242 ZW$=HA$(ZU):ZW=HA(ZU):HA$(ZU)=HA$(ZV)
6244 HA(ZU)=HA(ZV):HA$(ZV)=ZW$:HA(ZV)=ZW
6246 ZW=HB(ZU):HB(ZU)=HB(ZV):HB(ZV)=ZW:RETURN
6250 Z2(0)=0:IFAS=3THENFORC=1TOHA%:Z1$(C)=HA$(C):Z2(C)=HA(C):Z2(0)=Z2(0)+HA(C):N
EXT
6255 IFAS=3THENZ3%=HA%:HA(0)=Z2(0):GOTO6290
6260 IFAS=4THENFORC=1TOSO%:Z1$(C)=SO$(C):Z2(C)=SO(C):Z2(0)=Z2(0)+SO(C):NEXT
6265 IFAS=4THENZ3%=SO%:SO(0)=Z2(0)
6289 REM RVS-GRN OFF-BLK
6290 FORC=1TOZ3%:IFZ3%=0THENPRINT"■ SORRY, NO INFORMATION STORED ■":RETURN
6295 IFC<10THENPRINT" ";
6300 PRINTCTAB(4)Z1$(C)TAB(28)"£";:Z9=Z2(C):GOSUB13750:NEXT
6310 PRINTTAB(25)"-------------"
6320 Z2(0)=INT(Z2(0)):PRINTTAB(26)"£";:Z9=Z2(0):GOSUB13749:RETURN
6499 REM CRD RED BLK
6500 AA=Z9:CF=0:PRINT"■ TO CEASE INPUT TYPE ■*■"
6505 IFAS=4THEN6550
```

```
6510 PRINT" ADD OR SUBTRACT (A/S)"
6520 GETAN$:IF(AN$="*")OR(AN$="N")THENRETURN
6530 CF=0:IFAN$="S"THENCF=1:PRINT" SUBTRACT";:GOTO6550
6533 IFAN$="P"ANDAS=3THENAA=AA+HB(CA):RETURN
6535 IFAN$="C"ANDAS=3THENCF=2:AA=0:PRINT" NEW ADVISED";:GOTO6550
6540 IFAN$<>"A"THEN6520
6545 PRINT" ADD";
6550 PRINTTAB(12)" AMOUNT £ ";:GOSUB13545
6554 REM RVS-RED OFF-BLK
6555 IFZA(NO)<0THENPRINT"■■ TOO MUCH■■":GOTO6505
6560 IFCF=1THENZA(NO)=-ZA(NO):CF=0
6570 AA=AA+ZA(NO):ZT$=STR$(AA):IFLEN(ZT$)>=ZXTHENAA=AA-ZA(NO):ZA(NO)=-1:GOTO6555
6575 IFCF=2THENRETURN
6580 GOTO6510
6749 REM CRD CRL
6750 PRINT"■ DETAILS (MAX"ZM"■) ";:CM$=""
6759 REM RVS OFF-CRL
6760 PRINT"■*■■";
6770 GETZ$:IFZ$=""THEN6770
6780 Z=ASC(Z$):IFZ>95THEN6770
6790 ZL=LEN(CM$):IFZL>(ZM-1)THEN6810
6800 IF(Z=32ANDZL)ORZ>32THENCM$=CM$+Z$:PRINTZ$;:GOTO6760
6810 IFZ=13ANDZLTHENPRINT" ":RETURN
6820 IFZ=20ANDZLTHENCM$=LEFT$(CM$,ZL-1):PRINTZ$;
6830 GOTO6760
7000 POKE53280,7:IFPEEK(808)<>225THEN20
7004 REM CLR CRD-RVS OFF-CRD
7005 PRINT"♥"TAB(12)"■■ ACCOUNTS MENU ■■":C=0:FORR=1TOR1%
7010 PRINTTAB(7)R1$(R);TAB(31)CHR$(64+R):C=C+1:IFC=3THENC=0:PRINT
7015 NEXT
7019 REM   2*CRD-RVS OFF
7020 PRINTTAB(7)R1$(0);TAB(31)"X":PRINTTAB(9)"■■■ PRESS RELEVANT KEY ■"
7030 GETAN$:IFAN$=""THEN7030
7040 AP=ASC(AN$)-64:IFAP=24THEN3000
7050 IFAP<1ORAP>R1%THEN7030
7060 Z6=FB(AP):PM=FA(AP)
7065 ONAPGOSUB7200,7300,7400,7500,7600,7700,7800,7900,8000,8100,8200,8300
7070 GOSUB9000:IFAQ=6THEN7060
7080 FB(AP)=Z6:PM=0
7085 ONAPGOSUB7250,7350,7450,7550,7650,7750,7850,7950,8050,8150,8250,8350
7090 GOTO7000
7200 Z3%=A3%:Z4%=A4%:FORC=0TOZ3%:Z1$(C)=A1$(C):Z2(C)=A2(C):NEXT:RETURN
7250 A3%=Z3%:FORC=0TOA3%:A1$(C)=Z1$(C):A2(C)=Z2(C):NEXT:RETURN
7300 Z3%=B3%:Z4%=B4%:FORC=0TOZ3%:Z1$(C)=B1$(C):Z2(C)=B2(C):NEXT:RETURN
7350 B3%=Z3%:FORC=0TOB3%:B1$(C)=Z1$(C):B2(C)=Z2(C):NEXT:RETURN
7400 Z3%=C3%:Z4%=C4%:FORC=0TOZ3%:Z1$(C)=C1$(C):Z2(C)=C2(C):NEXT:RETURN
7450 C3%=Z3%:FORC=0TOC3%:C1$(C)=Z1$(C):C2(C)=Z2(C):NEXT:RETURN
7500 Z3%=D3%:Z4%=D4%:FORC=0TOZ3%:Z1$(C)=D1$(C):Z2(C)=D2(C):NEXT:RETURN
7550 D3%=Z3%:FORC=0TOD3%:D1$(C)=Z1$(C):D2(C)=Z2(C):NEXT:RETURN
7600 Z3%=E3%:Z4%=E4%:FORC=0TOZ3%:Z1$(C)=E1$(C):Z2(C)=E2(C):NEXT:RETURN
7650 E3%=Z3%:FORC=0TOE3%:E1$(C)=Z1$(C):E2(C)=Z2(C):NEXT:RETURN
7700 Z3%=F3%:Z4%=F4%:FORC=0TOZ3%:Z1$(C)=F1$(C):Z2(C)=F2(C):NEXT:RETURN
7750 F3%=Z3%:FORC=0TOF3%:F1$(C)=Z1$(C):F2(C)=Z2(C):NEXT:RETURN
7800 Z3%=G3%:Z4%=G4%:FORC=0TOZ3%:Z1$(C)=G1$(C):Z2(C)=G2(C):NEXT:RETURN
7850 G3%=Z3%:FORC=0TOG3%:G1$(C)=Z1$(C):G2(C)=Z2(C):NEXT:RETURN
7900 Z3%=H3%:Z4%=H4%:FORC=0TOZ3%:Z1$(C)=H1$(C):Z2(C)=H2(C):NEXT:RETURN
7950 H3%=Z3%:FORC=0TOH3%:H1$(C)=Z1$(C):H2(C)=Z2(C):NEXT:RETURN
8000 Z3%=I3%:Z4%=I4%:FORC=0TOZ3%:Z1$(C)=I1$(C):Z2(C)=I2(C):NEXT:RETURN
8050 I3%=Z3%:FORC=0TOI3%:I1$(C)=Z1$(C):I2(C)=Z2(C):NEXT:RETURN
8100 Z3%=J3%:Z4%=J4%:FORC=0TOZ3%:Z1$(C)=J1$(C):Z2(C)=J2(C):NEXT:RETURN
8150 J3%=Z3%:FORC=0TOJ3%:J1$(C)=Z1$(C):J2(C)=Z2(C):NEXT:RETURN
8200 Z3%=K3%:Z4%=K4%:FORC=0TOZ3%:Z1$(C)=K1$(C):Z2(C)=K2(C):NEXT:RETURN
8250 K3%=Z3%:FORC=0TOK3%:K1$(C)=Z1$(C):K2(C)=Z2(C):NEXT:RETURN
8300 Z3%=L3%:Z4%=L4%:FORC=0TOZ3%:Z1$(C)=L1$(C):Z2(C)=L2(C):NEXT:RETURN
8350 L3%=Z3%:FORC=0TOL3%:L1$(C)=Z1$(C):L2(C)=Z2(C):NEXT:RETURN
8499 REM CLR RVS-BLU OFF-BLK-CRD
8500 PRINT"♥"TAB(10)"■■ ACCOUNTS MENU ■■■"
8510 FORC=1TOR1%:IFC<10THENPRINT" ";
8515 PRINTCTAB(6)R1$(C);TAB(30)CHR$(64+C):NEXT
8519 REM CRD RVS-RED BLK-OFF RVS-RED BLK-OFF RVS-RED BLK-OFF RVS-RED BLK-OFF
8520 PRINT"■ INSERT=■■1■■ AMEND=■■2■■ DELETE=■■3■■ EXIT=■■4■■
8524 REM RVS-BWN BLK-OFF
8525 PRINTTAB(19)"■■ ENTER CHOICE ■■":NO$=""
8530 GETAN$:AN=VAL(AN$):IFAN<1ORAN>5GOTO8530
8540 IFAN=4ORAN=5THENRETURN
8550 ONANGOSUB6093,6093,6096
8560 IFAN=1THENR1%=R1%+1:R2%=R1%:GOTO8590
8570 INPUT" NUMBER PLEASE";NO$:NO=ABS(VAL(NO$)):IFNO=0THEN8500
8575 R2%=NO
8580 IFNO<2ORNO>R1%THENPRINTTAB(16)M$(4):GOTO8570
8590 ONANGOSUB8800,8800,8900
8595 GOTO8500
8799 REM RVS-BLU BLK-OFF
8800 IFR1%=13THENR1%=R1%-1:PRINT"■■ SORRY, FILE FULL ■■":WAIT197,64:WAIT197,64,6
4:GOTO8870
8810 ZM=23:CM$="":GOSUB6750:R1$(R2%)=CM$:POKE198,0
8819 REM BWN RED BLK
```

Gas Electric Rates

```
8820 PRINT"■ INDICATE TYPE OF ACCOUNT :-":PRINT" ■0■ ORDINARY ACCOUNT"
8829 REM RED BLK RED BLK
8830 PRINT" ■1■ BANK ACCOUNT":PRINT" ■2■ CREDIT CARD ACCOUNT"
8840 GETAN$:IFAN$=""THEN8840
8850 AN=VAL(AN$):IFAN<0ORAN>2THEN8840
8860 FA(R2%)=AN
8870 POKE198,0:RETURN
8900 R1$(R2%)=""
8910 IFR2%=R1%+1THENONNO-1GOTO8925,8930,8935,8940,8945,8950,8955,8960,8965,8970,
8975
8920 R1$(R2%)=R1$(R2%+1):R2%=R2%+1:GOTO8910
8925 B3%=C3%:B4%=C4%:FORC=1TOB3%:B1$(C)==C1$(C):B2(C):C2(C):NEXTGOSUB8990:IFC=-1
THEN8980
8930 C3%=D3%:C4%=D4%:FORC=1TOC3%:C1$(C)=D1$(C):C2(C)=D2(C):NEXT:GOSUB8990:IFC=-1
THEN8980
8935 D3%=E3%:D4%=E4%:FORC=1TOD3%:D1$(C)=E1$(C):D2(C)=E2(C):NEXT:GOSUB8990:IFC=-1
THEN8980
8940 E3%=F3%:E4%=F4%:FORC=1TOE3%:E1$(C)=F1$(C):E2(C)=F2(C):NEXT:GOSUB8990:IFC=-1
THEN8980
8945 F3%=G3%:F4%=G4%:FORC=1TOF3%:F1$(C)=G1$(C):F2(C)=G2(C):NEXT:GOSUB8990:IFC=-1
THEN8980
8950 G3%=H3%:G4%=H4%:FORC=1TOG3%:G1$(C)=H1$(C):G2(C)=H2(C):NEXT:GOSUB8990:IFC=-1
THEN8980
8955 H3%=I3%:H4%=I4%:FORC=1TOH3%:H1$(C)=I1$(C):H2(C)=I2(C):NEXT:GOSUB8990:IFC=-1
THEN8980
8960 I3%=J3%:I4%=J4%:FORC=1TOI3%:I1$(C)=J1$(C):I2(C)=J2(C):NEXT:GOSUB8990:IFC=-1
THEN8980
8965 J3%=K3%:J4%=K4%:FORC=1TOJ3%:J1$(C)=K1$(C):J2(C)=K2(C):NEXT:GOSUB8990:IFC=-1
THEN8980
8970 K3%=L3%:K4%=L4%:FORC=1TOK3%:K1$(C)=L1$(C):K2(C)=L2(C):NEXT:GOSUB8990:IFC=-1
THEN8980
8975 L3%=0:L4%=0
8980 FORC=NO TO12:FA(C)=FA(C+1):FB(C)=FB(C+1):NEXT:R1%=R1%-1:RETURN
8985 RETURN
8990 NO=NO+1:IFNO=R1%THENC=-1
8995 RETURN
9000 Z1$(0)="BALANCE B/F":GOSUB14000:GOSUB11500
9004 REM RVS-BWN OFF-BLK
9005 PRINT"■■ ENTER CHOICE ■■":NO$="":ZU$=""
9006 REM RVS-RED BLK-OFF RVS-RED BLK-OFF RVS-RED BLK-OFF RVS-RED BLK-OFF
9007 PRINT" INSERT=■■1■■ AMEND=■■2■■ DELETE=■■3■■ RENUMBER=■■4■■
9007 REM RVS-RED BLK-OFF RVS-RED OFF-BLK RED BLK
9008 PRINT" EXIT  =■■5■■ ERROR=■■6■■";:IFC-1<>Z3%THENPRINTTAB(26)"NEXT PAGE=■*■"
;
9009 PRINT
9010 GETAQ$:IFAQ$=""THEN9010
9020 IFAQ$="*"GOTO9130
9030 AQ=VAL(AQ$):IFAQ<1ORAQ>6GOTO9010
9040 IFAQ=5THENZ5%=0:GOTO9170
9045 IFAQ=6THENZ5%=0:GOTO9180
9050 ONAQGOSUB9150,9150,9160,9160
9060 IFAQ=1THENZ3%=Z3%+1:Z8%=Z3%:DI=Z3%:GOTO9110
9070 IFAQ=4THEN9110
9079 REM CRD
9080 INPUT"■ NUMBER PLEASE";NO$:NO=ABS(VAL(NO$)):IFNO=0THEN9000
9090 IFNO<1ORNO>Z3%THENPRINTTAB(16)M$(4):GOTO9080
9099 REM   2*CRU CRU
9100 Z8%=NO:DI=NO:PRINT"□□                                □"
9110 ONAQGOSUB10000,10000,10100,10200
9120 GOTO9000
9130 IFC-1=Z3%THENZ5%=0:GOTO9000
9140 Z5%=Z5%+12:GOTO9000
9149 REM   2*CRU RVS-GRN CRL BLK-OFF-CRD
9150 PRINT"□□"TAB(AQ*8-1);"■■"AQ"■ ■■■":RETURN
9159 REM   2*CRU RVS-GRN CRL BLK-OFF-CRD
9160 PRINT"□□"TAB(AQ*11-9)"■■"AQ"■ ■■■":RETURN
9170 IFPM>0THENZ6=0:FORC=0TOZ3%:Z6=INT(Z6+Z2(C)):NEXT
9180 RETURN
10000 IFZ3%=Z4%THENGOSUB12000
10010 GOSUB10300:ZM=24:CM$="":GOSUB6760:Z1$(Z8%)=CM$:IFZ8%=<Z5%THENZ5%=0
10020 GOSUB13500
10030 RETURN
10100 IFPM=1THENZ2(0)=INT(Z2(0)+Z2(Z8%))
10110 Z1$(Z8%)="":Z2(Z8%)=0:IFZ8%=<Z5%THENZ5%=0
10120 IFZ8%=Z3%+1THEN10140
10130 Z1$(Z8%)=Z1$(Z8%+1):Z2(Z8%)=Z2(Z8%+1):Z8%=Z8%+1:GOTO10120
10140 Z3%=Z3%-1:IFZ5%=>Z3%THENZ5%=Z5%-12
10150 RETURN
10199 REM CRD RED BLK RED BLK
10200 INPUT"■ THE TWO NUMBERS ARE ( ■*■ , ■*■ ) ";ZU$,ZV$
10205 ZU=ABS(VAL(ZU$)):ZV=ABS(VAL(ZV$)):IFZU=0ORZV=0THENRETURN
10209 REM RVS-RED OFF-BLK
10210 IFZU<1ORZV<1ORZU>Z3%ORZV>Z3%THENPRINT"■■ INCORRECT NUMBER ■■":GOTO10200
10220 ZW$=Z1$(ZU):ZW=Z2(ZU):Z1$(ZU)=Z1$(ZV)
10230 Z2(ZU)=Z2(ZV):Z1$(ZV)=ZW$:Z2(ZV)=ZW:IFZU=<Z5%ORZV=<Z5%THENZ5%=0
10240 RETURN
```

```
10300 IFPM=1GOTO10400
10309 REM CRD
10310 PRINT"■ NO DATE   DETAIL      "TAB(31)"AMOUNT"
10320 PRINT:IFDI<10THENPRINT" ";
10329 REM CRD CRU   4*CRR-CRU
10330 PRINTDI"■───── ───────────────□":PRINT"■■■■□";
10340 GOTO10430
10399 REM CRD
10400 PRINT"■ NO DATE   NUMBER DETAIL"TAB(31)"AMOUNT"
10410 PRINT:IFDI<10THENPRINT" ";
10419 REM CRD CRU   4*CRR-CRU
10420 PRINTDI"■───── ───── ─────────□":PRINT"■■■■□";
10430 RETURN
11500 IFPM=2ANDZ3%=0THEN11590
11502 IFPM=2ANDZ5%=0THENZ1=0
11503 IFPM=1ANDZ5%=0THENPRINTTAB(4)Z1$(0)TAB(26)"£";:Z9=Z2(0):Z1=Z9:GOSUB13749
11505 IFPM>0ANDAQ$="*"ANDZ5%<>0THENPRINTTAB(4)Z1$(0)TAB(26)"£";:Z1=Z7:Z9=Z1:GOSU
B13749:GOTO11510
11508 IFPM>0ANDZ5%<>0THENPRINTTAB(4)Z1$(0)TAB(26)"£";:Z9=Z1:GOSUB13749
11510 CH=Z5%+12:IFCH>Z3%THENCH=Z3%
11520 FORC=Z5%+1TOCH
11530 IFZ3%=0GOTO11590
11535 IFC<10THENPRINT" ";
11540 PRINTCTAB(4)Z1$(C)TAB(28)"£";:Z9=Z2(C)
11544 REM BLK
11545 GOSUB13750:PRINT"■";:IFC=Z3%THENC=C+1:GOTO11555
11550 NEXTC
11555 IFPM=0THEN11590
11560 Z7=Z1:FORCT=Z5%+1TOCH:Z7=Z7+Z2(CT):NEXT
11569 REM CRL
11570 PRINTTAB(26)"───────────■":Z9=Z7:PRINT"  TOTAL OF PAGE"TAB(26)"£";
11580 GOSUB13749
11590 RETURN
12000 FORC=3TOZ3%:IFPM=1THENZ2(0)=Z2(0)+Z2(1)+Z2(2)
12010 Z1$(C-2)=Z1$(C):Z2(C-2)=Z2(C):NEXT
12020 Z3%=Z3%-3:IFPM<>0THENGOSUB9170
12030 RETURN
13499 REM CRU
13500 PRINTTAB(29)"□£";:IFPM<>1THENCS=1:GOTO13550
13509 REM CRU
13510 PRINT"□"TAB(30)"(+ OR -)"
13520 GETAN$:IFAN$=""THEN13520
13529 REM CRU-RVS-RED OFF
13530 IFAN$="-"THENCS=2:PRINTTAB(30)"□■■-■        ":GOTO13550
13535 IFAN$<>"+"THEN13520
13539 REM CRU-RVS OFF
13540 CS=1:PRINTTAB(30)"□■+■        ":GOTO13550
13545 ZA(NO)=0:ZL=0:ZN=0:CM$="":ZM=5:GOTO13560
13550 PRINTTAB(30);:CM$="":ZM=5:ZL=0:ZN=0
13559 REM RVS OFF-CRL
13560 PRINT"■*■■";
13570 GETZ$:IFZ$=""THEN13570
13575 ZL=LEN(CM$)
13580 IFZN>0THENZN=ZN+1:GOTO13600
13585 IFZ$=","ORZ$="/"THENZ$="."
13590 IFZ$="."THENZM=ZL+3:ZN=ZN+1
13600 Z=ASC(Z$):IFZ>57THEN13570
13605 IFZL=ZMGOTO13620
13610 IFZ>45ANDZN<2THENCM$=CM$+Z$:PRINTZ$;:GOTO13560
13615 IFZ>47THENCM$=CM$+Z$:PRINTZ$;:GOTO13560
13620 IFZ=13ANDZLTHEN13650
13625 IFZ=20ANDZLTHENCM$=LEFT$(CM$,ZL-1):PRINTZ$;:GOTO13635
13630 GOTO13560
13635 FORC=1TOZL:CQ$=MID$(CM$,C,1):IFCQ$="."THENZN=1:ZM=C+2:GOTO13560
13640 NEXT:ZN=0:ZM=5:GOTO13560
13650 IFCS=0THENZA(NO)=VAL(CM$)*100:GOTO13670
13660 IFCS=1THENZ2(Z8%)=VAL(CM$)*100:GOTO13670
13665 Z2(Z8%)=-(VAL(CM$)*100)
13669 REM BLK
13670 PRINT"■ ":CS=0:RETURN
13749 ZX=12:ZX$="         0.00":GOTO13753
13750 ZX=10:ZX$="       0.00"
13753 IFZ9=0THENPRINTZX$;:GOTO13790
13754 REM RED
13755 IFZ9<0THENPRINT"■";:Z9=-Z9
13760 ZA$=STR$(Z9):ZZ=LEN(ZA$):ZY=0:IFZZ<4THENZZ=ZZ-1
13770 IFZZ<ZXTHENZY=ZX-ZZ
13775 PRINTLEFT$(ZX$,ZY);:IFZ9<100THENPRINTRIGHT$(ZA$,ZZ);:GOTO13790
13785 PRINTMID$(ZA$,2,ZZ-3)+"."+RIGHT$(ZA$,2);
13789 REM CRL-BLK
13790 PRINT"■■":RETURN
14000 IFPM<>0THENGOSUB9170
14004 REM CLR RVS-BLU BLK-OFF
14005 PRINT"□"TAB(10)"■■ "R1$(AP)" ■■";:IFPM=0THENPRINT:GOTO14020
14010 PRINTTAB(26)"£";:Z9=Z6:GOSUB13749
14020 PRINT:RETURN
```



Adventures can be created from literary sources in Runecaster's book.

ALTHOUGH MANY DEDICATED Commodore 64 adventurers buy and play this type of software almost exclusively, some more hardy souls even write their own. The easy way is to use a program such as 'The Quill' by Gilsoft or its American version 'Adventure Writer' from the CodeWriter Corporation. Both of these programs work very well and are fairly easy to understand and master. A good number of fine adventures are already on the market, having been written using these utilities.

At present, you can only design graphics around those graphics characters accessed by the Commodore Shift key (the business character set – the left hand characters shown on your Commodore 64 keyboard) but Gilsoft have already bought out an add-on – The Illustrator – for their Spectrum Quill, so let's keep our fingers crossed for something similar on the Commodore 64.

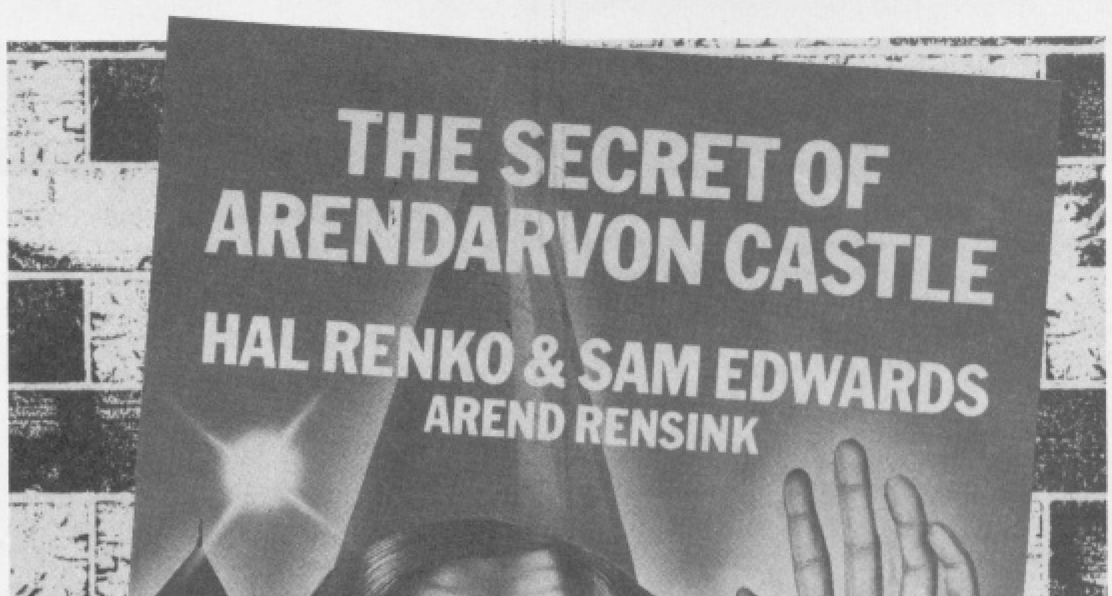
Even with the limited graphics available, I have seen some very acceptable results – just plan your pictures carefully, and before doing anything else, check out the graphics symbols you have at your command. There are one or two useful ones that are shown on the keyboard.

By the book

The next step down on the convenience ladder is to copy programs from either books or magazines. There is one rather different offering from now on the market from publishers Addison-Wesley of Wokingham called "The Secret of Arendarvon Castle". This is a 100 odd page book which costs £5.95 and not only includes the program to type in but also some 90 or so pages of history, clues and suggestions in a very novel form.

The basic storyline is that you follow in the footsteps of a journalist who has vanished whilst investigating a series of mysterious events that have taken place at this rather mystical castle in the far north west of Scotland. Arendarvon Castle has a long history stretching back into the Middle Ages and time and again it has been linked not only with events recorded in the history books but also a number of darker 'happenings' where alchemy and its associated magic seems not unlikely ...

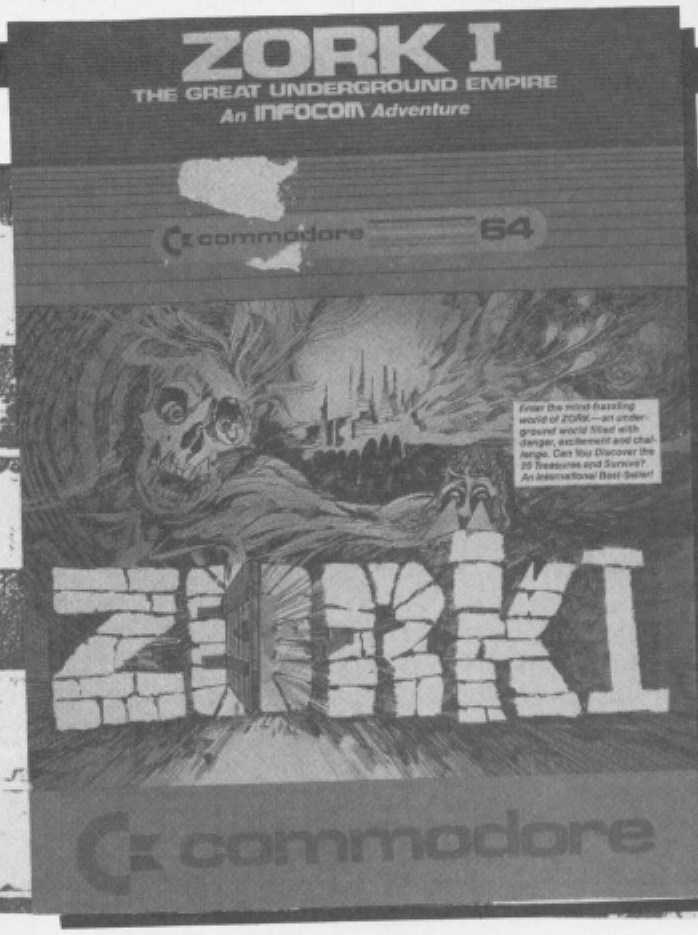
The part of the book that introduces, sets the scene and generally scatters what *might* be clues, is good reading and is set out in a very attractive style. Typefaces alter throughout the book to simulate the



journalists' typewritten script; newsprint from his clippings, collected during his research – an illustrated guide to the castle is reproduced, as are various maps, drawings and pictures. There are even reprints from the 'Magazine of the Supernatural', vol. XXVIII, no. 3! In fact, a host of goodies for you to study as you lie in bed planning your next moves on your future sessions with your Commodore 64.

The program itself is quite unintelligible as you type it in! There are several BASIC programs that enable you





to either create data files or access this data once you have finished typing it all in. The program uses a specially constructed language (?) called ALADIN, which involves *you* in typing in copious amounts of DATA lines.

Typing in such vast amounts of unassociated numbers and letters is just asking for trouble ... but, fear not, the original programmers have thought carefully about this topic and the BASIC program used to file away all this data has an error checking routine built in. This will draw your attention to the area where you have made a mistake! Failing that – you can buy the program on tape from Addison-Wesley.

Typing in a program of this size may well be a daunting task, but again the book presents us with an attractive (well acceptable anyway...) system to make this chore seem easier. The entire program has been divided into seven steps – it is suggested that you start on Monday and work your way through, using the given daily schedule! No rest on Sunday here...

Each day's input is checked as it is finished so all should be set for getting your game up and running sometime Sunday afternoon! Assembling the final tape takes a fair bit of time and, when you eventually RUN your adventure it does take a long time to load all the data.

I think that, with the care that has gone into the production of this system, one additional program might have been included to enable the loading of not only the master BASIC operating system but also that block of memory used to hold all the games data. This should not be a difficult task and would probably save over 15 minutes of reading the data from the sequential file created on tape.

The game itself is not outstanding but will keep your brain cells active for some time, whilst you sort out how to attempt a solution! 'Arendarvon' is text only and the response time to input commands is a little slow – but still not as slow as some games on the market. The program does not specify either screen or text colours and is displayed in upper case characters. So before you start, select colours to suit you and your television/monitor. Better still, write these lines in at the beginning of the final program. As for the text, I prefer lower case, but try both and choose which you prefer.

The game follows a fairly normal pattern with verb/noun input and the usual vocabulary of LOOK, EXAMINE, INVENTORY, etc.. The input commands may be expanded to give greater control and some examples given are: GO TO THE DOOR, OPEN YELLOW DOOR, PUT FLOWER INTO VASE and TAKE HOME SPELL. Note the use of TAKE instead of the more familiar GET also the somewhat unusual directional commands: GO LEFT, AHEAD, TURN RIGHT, etc. I found these a little awkward to start with, but very quickly got the feel for them after I had bumped into a few walls!

There are a number of spells you may use (once you have found them!) and there is also a MEASURE command to enable you to map out the rooms to see if there are any secret passages! You may SAVE your present position within the castle and RESTORE it at a later date.

All in all, there are quite a few novel concepts in 'The Secret of Arendarvon Castle'. I look forward to some more elegant programs from the same source in the future.

Another magic castle ...

One of Scott Adams' latest adventures takes you to a castle in a different realm altogether. 'The Sorcerer of Claymorgue Castle', distributed by Adventure International of Birmingham and available for the Commodore 64 on cassette and disc. It is a text and graphics adventure and the Commodore 'Sorcerer' has been implemented by Brian Howath of Mysterious Adventures fame. The name Scott Adams almost guarantees that you have a game to get your teeth into and that your brain will have to work overtime – many of his adventures make one aware of just how devious 'lateral thinking' can be when applied to this sort of product! In brief, Solon the Master Wizard lost 'the 13 Stars of Power' to another wizard, Vileroth. This happened a long time ago and Vileroth was unable to master the Stars' power. They proved his downfall but before he died he carefully hid the Stars. Now, after his destruction, Solon has sent his faithful young apprentice Beanwick (you) to try and recover these valubale artifacts.

The graphics are good and take little time to draw onto the screen. The location text descriptions are kept to a minimum and, if you change what would be seen, a new graphic is produced.

The number of locations which you can visit initially is fairly limited and you are, thereby, forced to find out how to use what you have got with you – namely the six spells your kindly master, Solon, has given you. In fact, you cannot get into the castle without using one of them.

The input command routine is strictly limited to two words – for normal purposes, verb-noun. The vocabulary is not enormous but reasonable synonyms are accepted. I feel that a certain amount of guidance should be given in the instructions – I prefer to get on with the game than wrestle with using the English language! A small point is that, when I cast the Fire Spell, the response was "OK in 2



words; at what?". It took a few frustrating minutes to learn that the first of the two required words was 'AT' ...simple once you know!

You may SAVE GAME at any point and carry on from there and RESTORE to the saved position on being killed, quitting or starting afresh. Although the instructions say you may save your position with a coded file name, this too does not appear to be found in the Commodore 64 tape version – no matter just make a note of the tape counter and set the tape position for the 'restoration' you require!

On first RUNning the program, you are asked: "User Default Colours?" This needs an input of 'Y' or 'N'. 'N' lets you set your choice of foreground (text) and background colours – I wish more programs offered this simple but very useful option.

For all the small frustrations, 'The Sorcerer of Claymorgue Castle' is an adventure that I find I have to return to. As yet, I have only located three of the Stars and seem to be baulked at finding many additional locations ... but give me time, I'll get there!

Evergreen classic

Until recently, I had not had the opportunity to try any of what have become recognised classics in adventure circles – The Zork Adventures.

Zork has quite a fascinating history and was originally written to run on a Digital Corporation PDP 10. It eventually grew to such a size that it evens stretched the megabyte capacity of that! All this took place in the pre-history of the micro computer as we know it – back in the 1970's!

Zork – as it was then – was shrunk somewhat to suit the meagre capacity of the 5¼ in. disc and became available for both the Apple II and the TRS 80 and, much later, the Commodore 64. Even with judicious shrinking Zork was too large to accommodate as one program and was split into separate self-contained adventures. These are linked by a common theme and Zorks I, II and III are now marketed under the Commodore label, although they still come under the original Infocom copyright.

Zork games are disc based and load in additional data from disc as the adventure progresses. Initially, you may wait a few seconds as it loads this data relevant to new locations but, data is accessed in relatively small doses so the occasional response lag is acceptably short. This system, of course, means that you have an adventure that, in size, greatly exceeds the available memory capacity of your computer. This in turn allows for the breadth of descriptive location texts that have influenced such British market leaders as Level Nine.

The marks of the Infocom adventure games of which the Zork trilogy are only a part (others for the Commodore 64 are 'Suspended', 'Starcross' and 'Deadline' – all marketed under the Commodore label) are the highly descriptive prose, at present text only, and the ability of the command interpreter to deal with what, for adventure games, is a relatively complex sentence, for example: DROP ALL BUT THE KNIFE AND THE LAMP. Although this undoubtedly makes many actions easier (TAKE ALL), it also could add greatly to the challenge of communication! The first time I played Zork I, I had some difficulty in passing the Troll – mainly because I was trying to be too clever and overlooked the more direct approach.

As with most adventures, mapping is particularly important with the Zork trilogy, especially in the initial stages in Zork I as you wander outside a boarded up white house. This is a maze of sorts and, although, you are unlikely to become irretrievably lost – use your pencil and paper.

These adventures display one attribute I've never been terribly keen on – what I call the 'twisty passage syndrome' – where going west after travelling east, all too often, does *not* return you to the original starting place! This trick does sometimes seem to be used for confusion's sake rather than for the game's sake.

If you have never tried Zork – at least chat up your local computer shop to run it up for you; it is a classic and the leaflet you can find at the first location in Zork I is quite right when it says "Welcome to Zork – No computer should be without one"!

Zork has a number of commands, such as VERBOSE – you get the full literary treatment with each location description. Alternatively, you can have BRIEF, which is the shortened form which you usually see on returning to a location you have already visited. Or, if you really want it short and sweet – SUPERBRIEF – where you just get a one or two word description of where you are. You may even pose certain questions by using WHERE IS ... or WHAT IS ... You may not always get a satisfying answer but it's nice to have the unusual option of being able to *ask* the computer something.

You may fight some of the creatures you meet on your travels although most, if not all, will fight back if they are able to. You may DIAGNOSE to check on your present health and this will also tell you how many turns you need to return to perfect fitness if you have been wounded!

Disc versions only for Zork but, if you have a disc unit, treat it to one of these ... it (and you) won't regret it!

Exmon
★ ★ ★ ★ ★
Interceptor
£7.00
CBM 64

THE EXMON MONITOR IS A MACHINE code utility for the Commodore which will allow you to enter, edit, debug, save and retrieve machine code programs with surprising ease and efficiency.

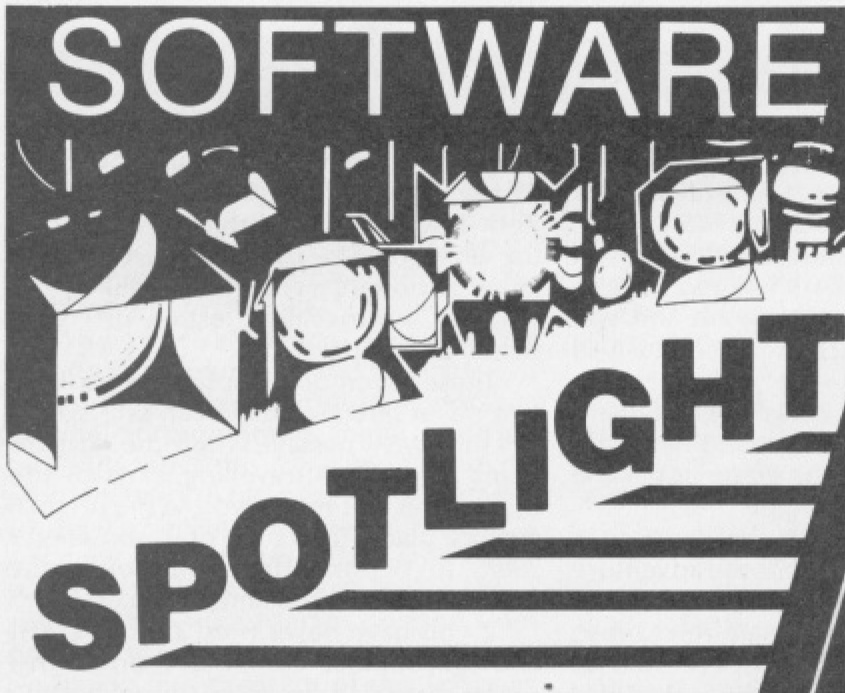
The accompanying six page manual, although concise in nature, is well written and clearly describes the various powerful commands and facilities that are available to you.

EXMON incorporates all the usual features, an easy to use assembler/di-assembler, extensive memory and internal register editing facilities along with a host of commands which will make light work of moving, saving, loading and verifying large chunks of code. An impressive array of I/O commands are available which will enable simple printer formatting, reading and writing to specific disc blocks along with other versatile disc and tape management facilities.

Apart from the usual facilities one would expect from a good monitor, this one includes several gems which make it stand out: for example, there is a 'HUNT' command which will search for a specified character string displaying its memory address when found. A calculator facility is another nice touch which will add and subtract hex numbers, convert hex numbers into decimal or vice versa, and even calculates hex offset values for the use of branching instructions all of which are most useful for those who, like me, have mental blackouts trying to manipulate hexadecimal numbers.

All in all EXMON is a quality piece of software which has been designed primarily to meet the needs and demands of the serious machine code programmer, yet retaining the ease of use which will enable the less experienced programmer to exercise far greater flexibility and control when using machine code.

P.C.





Our reviewers are here again to throw some light on the new software releases.

Jumpman
★ ★ ★ ★ ★
CBS Software
£8.95 (cassette) £11.95 (disc)
CBM 64 + Joystick

NOW THIS IS WHAT I WOULD CALL AN addictive game. Jumpman is another game from the Epyx Stable. The plot is that on Jupiter the space-station has been infiltrated by some 'not so friendly' undesirables. They have travelled through all 30 levels planting bombs and generally being a nuisance. You (Jumpman) are the government's top secret weapon and it is up to you to stop the station from crumbling any further.

The menu allows you to choose which game variation you wish your Jumpman to start at. It also allows you to choose how many players are going to try to save the station. It is possible to select the speed of your Jumpman before you enter a level. The game levels are split into 3 sets, the beginners, the intermediate, and the advanced levels. As with one of Epyx's other games, Breakdance, you can select the grand loop which will take you through all the 30 levels (if you make it), or they have another category called Randomizer which will randomly choose a screen for you to play. If your Jumpman achieves a high-score, then it is recorded along with any bonus that you might have picked up on the way. When playing the game I found that when I jumped from one place to another (instead of falling to my death), if I only touched the corner of it, I could hold onto the ladder or walkway by pushing my joystick up. The other obstacles to keep in mind are the bullets which just float across the screen until they are in line with you and on other screens, robots or vampire bats! I thought it was an easy game to grasp and the graphics/sound were good.

S.L.F.P.

Raid on Bungeling Bay

★ ★ ★ ★
Ariolasoft
£9.95 (cassette); £12.95 (disc)
CBM 64 (joystick)

For those of us not fortunate enough to be invited to the launch of Ariolasoft at the Hippodrome in London, there still remains the pleasure of purchasing their games. Raid on Bungeling Bay falls into the second category.

The plot you may have heard before, you are the last resort etc etc, and must pilot a powerful and highly manoeuverable helicraft and bomb and destroy six factories which are scattered throughout an Island complex. You start out with 5 lives plus one aircraft carrier from which you operate, lose this and the game becomes very hard indeed, as you then have nowhere to repair or re-arm.

As you eventually destroy the factories the opposition hots up, there is only one game level but as you get better the game gets tougher. Defending the enemy positions against your attack are robot tanks, anti-aircraft guns, fighter aircraft and bombers – and while you're off bombing and wreaking havoc the enemy is secretly building a battle ship and trying to sink your Aircraft carrier, all good clean fun!

The main battle area is spread over 100 screens which feature wrap around, so if you fly off the left hand edge you will re-appear on the right hand side, with a similar effect operating north and south. You view the whole game from above your helicraft and, to complete the display, there is the mandatory radar screen with the accompanying score and lives remaining running across the bottom.

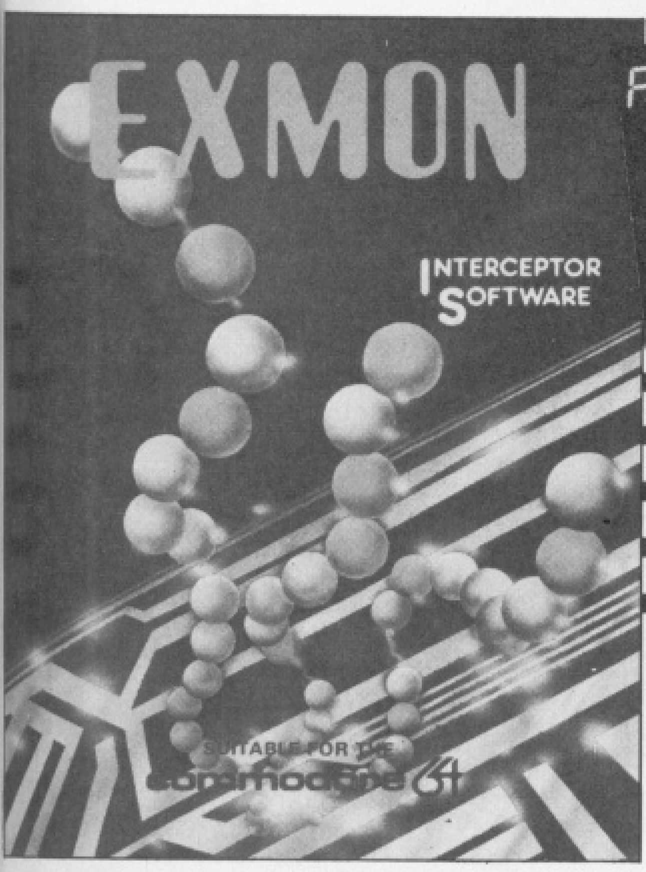
Graphically the game is superb with great detail paid to everything, brilliant buildings and airstrips on the islands and silky smooth scrolling in every direction. The sound is of the usual 'zap pow' variety found in this type of game, nothing new, just well done.

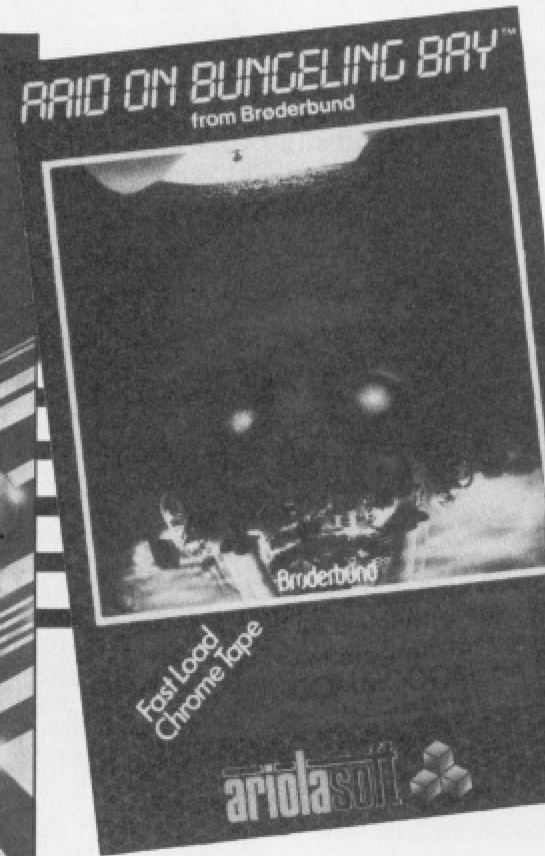
In play, your helicraft takes some getting used to but can be controlled from either keyboard or joystick, my own preference being a combination of both, direction via the joystick and weapon control from the keyboard.

So far my best effort is 4 factories destroyed but even with this I still didn't manage to catch sight of the high score table that the blurb tells you will appear if your score is high enough, whether this is a bug or just my poor scoring only further play will tell.

My review copy arrived on disc but I'm reliably informed that a turbo load tape will be available when the game is released early in '85. If, like me, you want more than space invaders or Manic Miners then this game has it all, a must for all wargamers, highly recommended.

M.T.U.







Cadcam Warrior

★ ★ ★ ★
Taskset
£9.95 (cassette) £12.95 (disc)
CBM 64 + Joystick

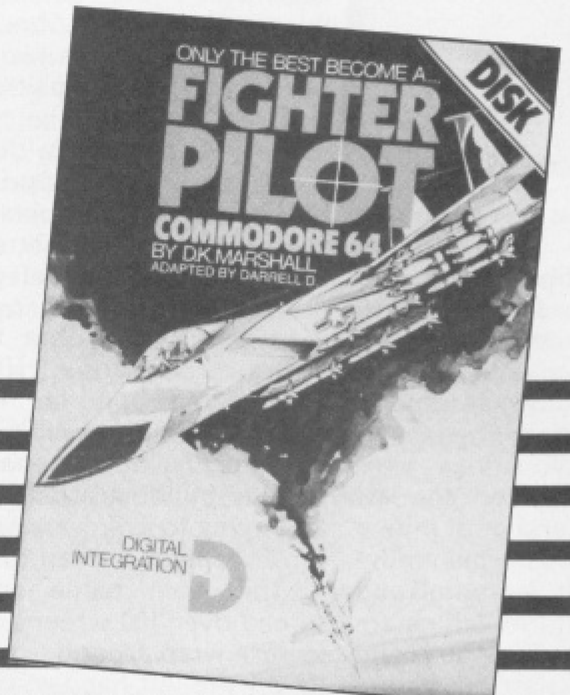
THE FIRST THING I NOTICED WITH CADcam Warrior was that it took longer to read the manual than it did to load the game! Whilst it was loading, (using a fastloader), the background to the game was revealed in a short story which was in the manual. You are Marsh and in control of an android called MAD-2. The task set for your droid is to push back into place a P.C.B. which has become dislodged inside the Enviro generator. The only problem is that the internal defence mechanisms are on guard to stop any unauthorised access (even by it's own droid). The only other obstacle in this game are the 8192 screens which could be used throughout.

The game is designed in such a way that if you clear a screen of enemies, a choice of screens is displayed. The manual suggests that while playing the game it is worth making a map. I would also suggest roping in a stray friend to draw a map while you go zapping every creature. The game is made easier with the inclusion of a colour coded strip on the side of every screen. This will indicate how to get through the particular screen on which you are operating. As if that is not enough, on various screens you will also find items which can be picked up; in some cases these will finish a screen.

Finally, the graphics on this game are up to the usual high standard that I have come to expect from Tasket; the sound is good and the game content enormous. But, I somehow feel that having 8192 different screens is just a little too much. Still you cannot complain about not getting value for money!

S.L.F.P.

SOFTWARE SPOTLIGHT



Fighter Pilot
★ ★ ★ ★
Digital Integration
£9.95 (cassette)/£14.95 (disc)
CBM 64 (keyboard and/or joystick)

AFTER A VERY SLOW START, FLIGHT simulator programs are appearing thick and fast for most micros, the Commodore 64 being no exception. One of the latest to appear is Fighter Pilot from Digital Integration. The program is claimed to be a 'real-time simulation of the F-15 Eagle' offering 'stunning 3D graphics, full aerobatic performance, air to air combat' and many other tempting simulator characteristics.

This all adds up to quite a claim so does the program live up to expectations? After the title page you are given a menu which gives you six flying options and a pilot rating. The 'landing practice' option is particularly useful as reuniting a hairy F-15 with the ground is not something that comes easily.

The best option though, and this is what the program is really all about, is the air-to-air combat option. You commence your mission from base by blasting off from the runway with the re-heat turned on and in no time you are at 30,000 feet and homing onto your first target. This is no problem as your radar and other instruments guide you to the quarry. Providing you are within a mile of the target aircraft and at roughly the same altitude the enemy will appear in the cockpit window and then you let him have it. I felt a slight tinge of remorse as the enemy aircraft looks very much like the Avro Vulcan, one of my all-time favourite aircraft. It's no good being nostalgic though as he packs a hefty wallop. If you don't get him he'll get you or go on to bomb all of your airfields so you will be unable to land again!

The problem with some flight simulator programs is that the aircraft response to the keyboard and joystick is decidedly sluggish. Not so with Fighter Pilot. The response is immediate and this adds to the realism of the program. Fine adjustments to the aircraft's attitude are easy and instantaneous. This is backed up by the graphics which really are excellent. Sound is adequate and is mostly limited to the sound of the jet engines and your cannons firing.

Inevitably, flight simulators demand the use of a large number of keys for the various aircraft functions, in addition to the UP, DOWN, LEFT, RIGHT movement of the joystick. Fighter Pilot is no exception although in flight only two or three are really needed in addition to the joystick. However, to get the full enjoyment ftom the program, as with all flight simulator-type programs, some knowledge of the principles of flight would be helpful and considerable practice is needed to master the various keys and their uses. Therefore, I doubt if this will appeal to the younger fraternity who have been weaned on shoot-em-ups. Nevertheless, it is well worth the effort in trying to master and it's one you'll return to more than most. Shame about the Vulcan though!

I used the disc version which took about 1½ minutes to load so, unless the cassette version is a 'turbo-load', it would probably take about 15 minutes to run-in! The disc came in a very nice protective wallet but it does cost £5 more than the cassette!

J.F.

Castle of Terror
★ ★ ★ ★
Melbourne House
£9.95
CBM 64

FROM THE SOFTWARE COMPANY THAT gave us 'The Hobbit' comes this graphical adventure. Castle of Terror is set in the 1800s, in a village near a large castle – home of a secretive Count; sounds familiar?

As you wander around the village, information concerning the Castle and its owner can be obtained by talking to the villagers, if you talk to them nicely. Various objects to help you in your quest can also be found in and around the village, but they are not always obvious.

The graphics are very good and are drawn quite quickly, 4 or 5 seconds in most cases, which is just as well as the pictures are drawn for each location you visit, be it for the 1st or 21st time. Throughout the adventure, suitably eerie music is played, changing to suit the mood of your present location. Many adventurers may find the continuous music off-putting or monotonous, but I felt that it complemented the game quite well.

Text input and acceptance is on a par with 'The Hobbit' and in general is quite user friendly. A 'help' command is available but so far I have not found any real help from this source. Time will tell.

The program loads in 4¼ minutes, but further sections have to be loaded as you progress through the adventure, no doubt owing to the complexity of the program. I can heartily recommend this to new or seasoned adventurers alike.

D.I.W.







Burnin Rubber
★ ★ ★
Audiogenic
£8.95
CBM 64

WHEN YOU'RE NOT SPENDing long hours slaving over a hot keyboard and joystick, do any of you out there go to the cinema? You do remember the cinema don't you? It used to be popular before computers. If you do remember then you might have seen Death Race 2000; Burnin Rubber from Audiogenic is the veritable game of the film.

According to Audiogenic you're an entrant in the Intergalactic four seasons death race with the sole objective of running as many other cars as possible off the track, (makes a change from Pole Position). There are various different types of other entrants, most of which can be dispensed with reasonably easily, all except the indestructible tank cars, these should be avoided!

To help you add to the carnage and get out of tricky situations, your super car is able to fly for a limited period when speeding in excess of 100MPH. I needed to do this on numerous occasions to avoid the tracks obstructions which appear from time to time, completing a round enables you to pass on to the next season. Scoring is by destroying everybody else on the track, the bigger the opposition the higher the score, with a bonus feature if you manage to complete a whole round. There are supposedly more than 32 different tracks to try out but only the first 16 available from the start.

The graphics are good without being brilliant: the flying car is particularly good. Sound and music are also done well with the added advantage that you can toggle the sound on and off with the f5 key. The car is controlled by keyboard or joystick with the latter being by far the best bet. A high score table is included for those who want to record their efforts, as is a two player option (one after another, not simultaneously).

My only moan with this game was the long wait between using the high score table and starting the next game, made worse by the terrible addictiveness of the game itself. I reviewed Burnin Rubber over the Christmas period and suffice to say that I was still trying to get a higher score than my brother at 1.30a.m. Boxing day morning. Definitely one for the collection.

M.T.U.

Space Shuttle (A journey into space)
★ ★ ★
Activision
£9.99
CBM 64 + joystick

ALTHOUGH THE PLAYER IS instructed to follow the on screen instructions, I had to refer to the manual since there was a distinct lack of 'on screen instructions'.

This 'Flight Simulator' has 3 levels. Level 1 is 'Flight # 1 Autosimulator', whereby you carry out an abbreviated space mission where most of the operations are computer controlled. Level 2 is 'Flight # 2 Simulator', whereby you have a little more control of the system and level 3 is 'Flight # 3 STS 101', which is a fully fledged Shuttle Flight.

You are supplied with a function key overlay card since the simulator requires use of the keyboard as well as the joystick. The joystick and fire button control many functions, including the movement in any direction in space. The keyboard takes care of functions such as cargo door operation, landing gear operation, engine cut off etc.

The simulation goes through many stages including lift off, stabilizing orbit, satellite docking, de-orbit, re-entry and landing.

The first thing you must do is to select which simulation flight you require (1, 2 or 3 as described above). The display screen throughout consists of the dash board controls to the bottom half of the screen with a view of the front outside of the shuttle through the windows above the controls. On lift off, the shuttle shakes under the force of the craft leaving the gravitational pull of the Earth and the view through the window changes colour until you have established orbit. In orbit, the deep blue of space is speckled with star constellations (many can be recognised), whilst the Earth scrolls just under the edge of your window. Space flight can be left, right (sorry, port or starboard), up, down, backwards or forwards.

When a Satellite is spotted, you must dock with it as many times as possible. Throughout all operations, you can see exactly where you are in relation to the satellite by way of reference to the tracking devices on your control panel.

The game was infuriating since I never managed on flight 3, to dock with the satellite or land back on Earth. I shall, however, return to this program time and time again until I have gained some degree of skill. As the manual states, it requires a lot of practice to complete a successful mission. But if it was too easy then I should have got bored with it from the start.

Overall the program had good graphics and sound, good use of the joystick and keyboard and a good manual (which must be studied whilst you are running the program for it to make sense). One useful facility is a pause key which freezes the program so that you may refer to the manual.

S.D.E.

David's Midnight Magic

★ ★
Ariolasoft
£9.95
CBM 64

QUITE FRANKLY I'M SURE DAVID CAN think of plenty more magical things to do around midnight than getting his computer to teach him how to become a pinball wizard. Surprise, surprise this is yet another electronic pinball game. At least it has the saving grace of sociability in allowing four players to join in. There's also a jostle facility which will turn into a tile if you use it too much. Did I hear you ask what a tilt is? Naturally there are loads of bonus points to be mopped up for hitting all the right bits and pieces and a collector in the top of the screen which hands out bonuses as well as giving you the chance of a three ball game if you get three in it. But, let's face it, it's nothing really grand and not a patch on the real McCoy.

K.M.



SOFTWARE SPOTLIGHT



Choplifter

★ ★ ★ ★ ★
Ariolasoft
£9.95 (cassette); £12.95 (disc)
CBM 64 + joystick

THE INITIAL SCREEN SHOWS a helicopter waiting at base on the launch pad with an American flag clearly flapping in the breeze on top of HQ. Pressing the fire button starts the action in which you have 3 lives (clever, because the guys on 'Airwolf' seem to have hundreds).

Your mission is to rescue hostages locked in barracks. There are 16 hostages for each hut and a total of 64 hostages in all. Your chopper can carry a maximum of 16 men. So, off you go into the wild blue yonder controlling your helicopter in all directions.

The first hut has been blown open for you and, as you fly across the countryside and over the huts, you can see the little hostages breaking free, running in the direction of the helicopter and waving to you. You must land and let these idiots on board as quickly as possible. They must be idiots, since whilst you are trying to land to rescue them, you have to contend with tanks, jet fighters and drone air mines (of the homing variety) and all these little men can do is run into the line of fire and get shot at in mid wave. The attacks from each of the enemy are progressive commencing with just the tanks, then tanks and jets and then the drone air mines as well. These little guys can certainly sprint to you when they want. If t hey are shot by the enemy or you unfortunately land on them, they become deceased hostages.

You can shoot down enemy aircraft and drone mines in mid flight, but to bomb the tanks you must first go into tank mode which is via a medium press of the joystick fire button. A short press on this shoots or drops bombs (when in tank mode) whilst a long press turns your chopper through 180 degrees.

When your helicopter is full of men, or when the need arises, you can return to HQ to drop them off so that you may go back to rescue more men. The instructions refer to HQ as the Post Office. Perhaps you are dropping these men off to buy some stamps or some 'Get Out Soon' cards for their, as yet unrescued, pals. Or is it that Post Offices hide a deeper secret in America than we realise?

The game looks simple, but is not. It is very addictive and incorporates a random facility of attack, i.e. you don't know when or where the enemy are coming from. The game can be quite different each time it is played and there is always that element of surprise against you. As your whirlbird's blades rotate, there are accompanying sound effects. It is very amusing when you drop men off at HQ and they wave goodbye to you. Perhaps the sound effects of a cheer or two at this stage (if it were possible) would have had me rolling about at these ant like figures antics.

The 3D type graphics are simple but extremely colourful and very good. I found the instructions on the packaging very clear and easy to follow and the game was a pleasure to get into. Definitely for all age groups. Practice makes perfect with this game to a certain extent, together with a small amount of luck. I did find that each time I played, I became involved in the action in as much as I was part of the screen. I was there with the men being rescued and their actions were realistic: they seemed to have minds of their own.

S.D.E.

Juice

★ ★ ★
System 3 Software
£7.95
CBM 64 + Joystick

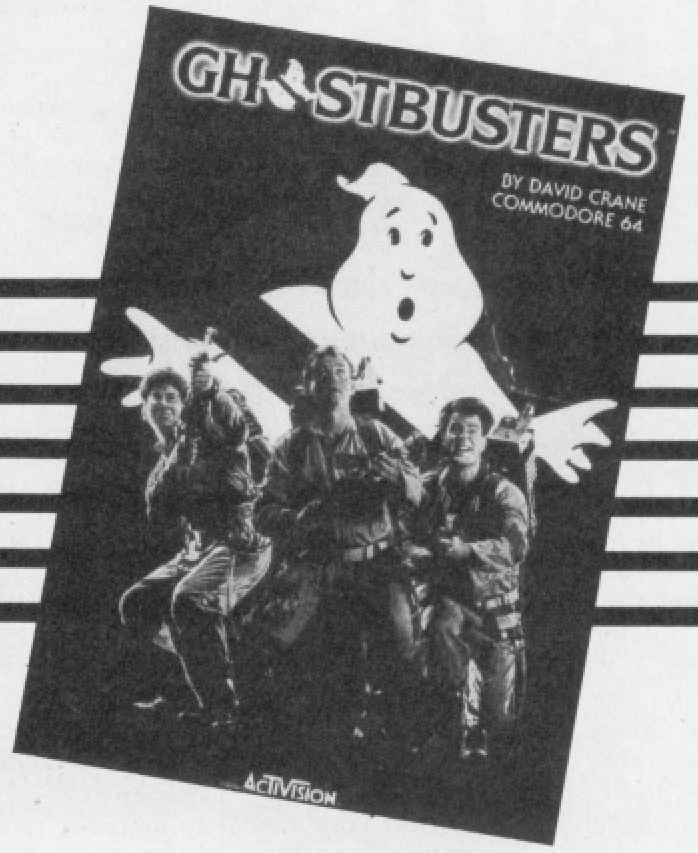
NOW I KNOW HOW THEY PRINT circuit boards! A little android called Edison jumps about the P.C.B. laying down the tracks for the electric current to run through, but he is being hampered by some shocking enemies. Coming back down to earth, this game is fairly original although I have seen a similar board layout on Flip & Flop. The idea is to connect up the whole of the P.C.B., thereby enabling the juice to flow. But, there are some other characters in this game who are out to stop you. The main opponent is Killerwatt, who will follow you around until either he gets you, or you make him fall off the circuit board by jumping onto a teleport pad; he,not being a very bright spark, will try to follow you.

There are other characters in this game such as Nohm and flash who will also do things to you if you're caught. The board works in such a way that when you tread on a square, a piece of circuitry appears. On the later levels Edison will have to tread on each square more times to make more circuits. At the beginning of each level a short demonstration is shown on the screen. The facilities supplied with the game include a level selector (up to level 6), a pause button, and game options which allow the selection of which enemies you require. if any.

The graphics are good although the sprite details could have been mutlicolour and the sound was a little too basic. But for me, the positives outweighed the negatives. It is also worth noting that there are bonus screens in the game which will boost your score by 2000 points if you manage to complete them by connecting one side of the circuit to the other in as short a time as possible.

S.L.F.P.





Ghostbusters

★ ★ ★ ★
Activision
£10.99 (tape)
CBM 64 + joystick

YOU'VE SEEN THE FILM, BOOK, POP video, LP and T-shirt – now play the computer game. Activision have been able to make the most of the film's current popularity by bringing their Ghostbusters game out simulataneously with the cinema release, but it would have been worth waiting for in any case.

The game is turbo-loaded in a remarkably short time considering the amount of data it must require. Once you've got it in, you're treated to a superb title screen featuring the Ghostbusters logo and a synthesised voice saying 'Ghostbusters', followed by an eerie, evil cackle of laughter. If you wait instead of pressing the start key, the program they plays the *entire* theme tune, in three voices, together with scrolling lyrics and a little singalong bouncing ball. This alone is almost worth the price of the game.

The actual game presumes that business is so good, the Ghostbusters are offering franchises. The bank lends you £10,000 and you start by spending it on your Ghostbusting equipment – choose carefully because you can't afford everything. Which is better, a Ghost Vacuum or a PK detector, a roomy slow estate car or a souped-up vehicle that carries less? How many Ghost traps should you purchase? Several games should help you decide the right priorities.

Once you're equipped, you drive around the city to haunted buildings attempting to manoeuvre the Slimers into your Ghost traps with your protom lances. (See the film – it'll all make sense then!) The faster you catch the ghosts, the more money you earn and the slower the city's PK (psychod-kinetic) energy level rises. Other ghosts are flocking to the Temple of Zuul, ever faster with rising PK, and towards the end of this phase you need to use your Ghost bait carefully to prevent the gian Stay-Puft Marshmallow man laying waste to city blocks.

If you complete the first phase earning more money than you started the game with, you pass to phase 2 where you have to sneak at least two of your men into the Temple of Zuul past Mr. Stay-Puft. I haven't managed to do this yet, but I'm told the ending echoes the film fairly faithfully.

The game is a lot of fun, and impressively programmed (the theme tune plays throughout, the graphics are good and there is another piece of synthesised speech). Remember that the realistic-sounding speech is done using only the basic 64 sound chip filters: no additional hardware is required. A round of applause for the programmers, please.

Recommended, even though you don't get a free Sigourney Weaver with every cassette.

P.G.

SOFTWARE SPOTLIGHT





Tim Love's Cricket
★ ★ ★
Peaksoft
£8.95
CBM 64 + 1 or 2 joysticks

IF YOU FANCY PLAYING A RELAXED game of cricket during those months of the year when the weather is doing it's utmost to re-create the ice age, then this program is for you. Simulating a 60 Over Match, it has all the features you would expect in a real match. Fast, medium and slow bowlers, scoreboard, bowling averages; it's all here. The game can be played by one or two players, but even at 'Village Green' level the computer proves to be a worthy opponent – rarely missing catches and never delivering a no-ball.

You can choose to name your own teams or play an England v The World match. Naming your own teams revealed a bug which prevented the teams having names longer than one character.

In play, the bowler is moved to left or right and the speed of his delivery varied. When he makes his delivery, the screen scrolls to bring the batsman into view. The batsman must be positioned to hit the ball, adjusting his bat to place the shot. Once the ball is hit (or missed) the view changes to an overhead picture of the field. The fielding team can freeze the action while a convenient fielder is chosen to intercept the ball.

The graphics are adequate although sound is virtually non-existent, limited to a simulation of the hallowed thwack of wood on leather. The pace of the game is steady rather than exhilerating, as would be expected. Unfortunately, the program adopts the same unhurried pace while loading, taking 9 minutes.

A good implementation not to be missed by cricket lovers, but probably too slow for arcade fans.

D.I.W.

Gandalf the Sorcerer
★ ★ ★
Tymac
£9.95 (cassette) £14.95 (disc)
CBM 64 — Joystick essential

GANDALF IS THE IRASCIBLE WIZARD IN 'The Hobbit' who has now escaped from both Tolkien and Melbourne House, and is busily defending his castle against marauding Lizardmen. These are miniature tyrannosauruses (or is it tyrannosauri?) who are trying to steal Gandalf's treasure. Fortunately he can shoot them with blasts of magic power and, as everyone knows, lizardmen turn into gold coins when they die. To score points, Gandalf needs to go outside and collect the coins before the yellow ostrich steals them.

This is the basic plot of the game, though it is complicated by the occasional appearance of some especially deadly spiders and the capture of Gandalf's apprentices. On screen two the apprentices mysteriously turn into Princesses – rather a silly programming error!

In both screens the graphics and animation are quite superb – amongst the best I have seen. The sound-effects are good, though the background tune is monotonous and irritating. There are no high-score tables or levels of difficulty.

I found the game interesting and amusing, though I wonder if there is enough variety to maintain interest for long. Nevertheless, it is very original and fun to play.

P.R.B.

Suicide Strike
★ ★ ★
System 3 Software
£7.95
CBM 64 + joystick

YOU ARE SITTING IN THE COCKPIT OF A plane with the horizon stretching out in front of you. Not a particularly pleasant sight considering you have been chosen for the suicide strike on an important enemy military target that must be destroyed at all costs. You're short on time and you're limited on fuel so you have to watch your speed carefully. The faster you fly the more fuel you use. And where would any zapping game be without some enemy hardware to blast away at? But with a slow firing gun and a potential for angling your shots, you've got to make every one count. First there are the busy little helicopters to destroy, then the hordes of sluggish tanks backed up by the hot-shot jet fighter squadrons before you reach your target, the radar stations. Seek out and destroy three of these in ten minutes and you will live to fight another day. To my mind its quite an effective game with reasonable 3D graphical effects. But then, give me a half-decent zapping game and I'm anybody's!

K.M.



Silicon Warrior

★
CBS Software
£8.95 (cassette); £11.95 (disc)
CBM 64 — Joystick essential

SUSPENDED IN SPACE IS A GRID OF twenty-five paving slabs with gaps between them. You can teleport from one slab to another, as in the USS Enterprise, and each slab you land on changes to your colour. Unfortunately, up to three opponents are busily teleporting about as well, changing them back again. One of these other players can be human, but otherwise they are computer controlled. Occasionally one of the slabs disappears, and if you fall down the hole you reappear in a tetrahedron-shaped cage. You needn't worry, though, as you are soon back on the pavement and can start teleporting again. The object of the game is to turn a row of five squares to your colour but, even if you succeed, your opponents can sabotage your efforts.

What a silly scenario, and what a boring game! Even if you add a little laser-fire, and call the slabs 'silicon chips' as in the eight page booklet supplied with the game, it remains flat and unimpressive.

Silicon Warrior is a variant of the well-known strategy game 'Connect 4' but the element of skill is largely destroyed by your opponents' unpredictability. It is a poor quality, over-priced import from America, and is not worth buying.

P.R.B.



One-on-one

★ ★ ★
Ariolasoft
£9.95 (cassette) £12.95 (disc)
CBM 64 + joystick

HERE WE ARE SUPPOSED TO HAVE TWO legendary greats of the basketball courts ready to battle it out in that general test of individual skill one-on-one. A variety of options allow you to select one of four skill levels: the player you want to play either against the computer or against somebody else and the length of the game either to a set score or to specified time limits. The computer characters' playing attributes are even supposed to reflect their real life skills. As in the real game you would expect fouls to be awarded. These are more rigidly enforced the greater the difficulty level chosen. You can even be hit by hot streaks when nothing seems to be able to go wrong and suffer from fatigue when anything can go wrong. Unfortunately, I found it a little disappointing primarily because of the fairly poor graphics and difficulty in telling what is actually going on. Skill in controlling the players with the joystick seemed to bare little relation to the result. Doubtless, if you are enamoured by basketball, then you'll probably be enamoured by this game.

K.M.

Snake Bite

★ ★
Firebird
£2.50
VIC 20 Unexpanded

SNAKE BITE IS BASED ON ONE OF THE earliest computer games written. I once played a version of this on an 8K Pet about four years ago. Those were the days.



This is a fairly good version of the game for the VIC. You control a snake which slithers about the screen eating various goodies. These come in the form of bugs, apples and grapes. As you eat, the snake grows. The longer it becomes, the harder it is to avoid its own tail. If it hits the tail the game is over. The game also ends if you hit a deadly mushroom or the screen border.

If you manage to clear a screen, more bugs and fruit appears. Your snake stays the same size and the game becomes very hard to play.

The game is not very original but is still quite playable. The graphics are clear and the movement of the snake is very convincing. Sound is fairly ordinary but adequate.

The most annoying feature of the game is having to type in the skill level at the start of each game. The screen prompt asks you for a number between one and five but only one, three and five are accepted.

As I said, it's not a bad version of the game. However, I've seen quite a few listings in magazines for this type of game, typing one in is probably better value than buying this one. It is a pity software houses can't think up and produce some new original games instead of churning out ancient copies of games like this.

P.R.

Fatty "Schizo" Henry

★ ★
Software Projects
£5.95
VIC 20 Unexpanded

A BIT SILLY THIS ONE. IT'S CERTAINLY an original idea for once. Let me explain the plot to you.

Apparently, based in the kitchen of a high class restaurant, weird things are happening. You take the part of little Twitchy Thwilly (?) the local octupus. You've been swimming about in your little aquarium when some weird customer decides to have you as the main course. Whisked out of the aquarium, you suddenly find yourself in an oven and this is where the game begins.

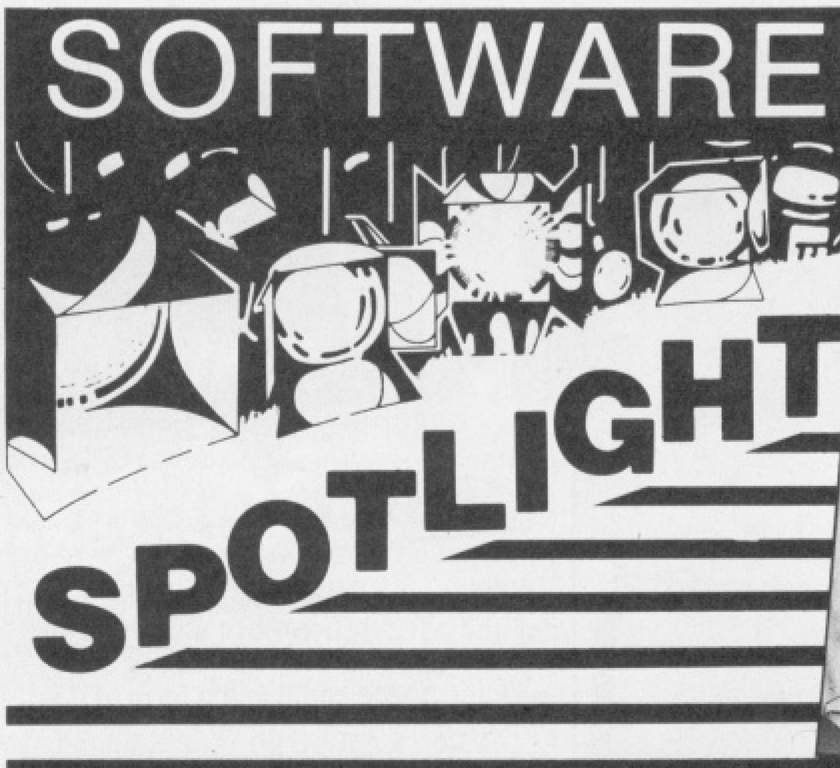
You must pick up a droplet of condensation from a pipe at the top of the screen. Then you have to take it down towards the flames at the bottom of the screen and drop it, putting out a flame. Once all the flames are out you get moved into a new oven where you have to repeat this process. There are ten different ovens in all and you can play any of the first five.

To stop you on your quest for survival are some strange nasties. These come in the form of food debris, oven foam and a rotating spit. Also to stop you is Fatty Henry the ugly thick dog who likes octupus for tea. But wait a moment, what the heck is that dog doing in the oven? Obviously a hot dog!!

It's not a bad idea for a game but it just does not seem to work. I found control of the octupus very difficult. The hazards were hard to avoid as all movement was very fast. Probably too fast, well for me anyway. Graphics are all in multicolour and frankly, pretty awful. Sound was nothing much either.

Although there are ten ovens they just don't seem to be much different. Still they are quite good for the unexpanded VIC. I wish Software Projects could have used a bit more memory and enhanced the game play. It's a good idea for a game but has not been written very well. I advise you to see it before you buy.

P.R.







Motocross

★ ★
System 3 Software
£7.95
CBM 64 + joystick

WHILST THE PROGRAM IS loading, you are treated to a title page display accompanied by the tune of Gilbert and Sullivan's 'A Policemans lot is not a happy one'. I do not know what the significance of this is, but it beats staring at a blank silent screen.

Once the program is loaded, you have to select your game and level of play. There are 3 levels to this game with each level having 2 games (i.e. a choice of 1 player or 2 players).

The display appears to be a desert with mountains and moving clouds in the distance and never alters throughout the game. The track scrolls towards you as you move and may twist and turn in either direction. The track border is made up of haystacks and is quite narrow. You may move the biker along the track only. The mountains never get any closer. A change of scenery or variation in track layout would have made this game more interesting.

At level 1 you have automatic transmission and no timer, whilst at level 2 you have a starter to wave a starting flag at you, a timer and manual transmission. Level 3 is the same as level 2, but with computer bikes to race against. You score 1000 points for every computer bike that you overtake, but lose 250 every time you are overtaken. The laps required to complete each level are 3, 5 and 7 respectively. Points are scored for every haystack passed but this varies depending on the gear you are in (transmission, not denims or leathers).

Whilst all this is going on, your bike's instruments are displayed at the top of the screen-speedometer, fuel gauge etc). I though this was a nice touch, but was so busy watching where I was going that I didn't have time to even have the occasional glance at the instruments. If I had, I wouldn't have been able to make use of the information, so it appears obvious that this has just been put on screen to fill up space and give a little more authenticity to the program.

The growl of the motorbike is good and changes as you change gear. Even the computer bikes are noisy and it is a different growl to your bike. Their noise varies as they pass you and as you pass them.

If you are a high score fanatic then this program is for you since practice will make perfect, but I found the game boring.

If you hit a haystack or a computer biker, you bounce across the track. You have to remount and continue. Perhaps the computer bikers have stabilizers fitted since they never fall off even when you crash into them.

Occasionally I got stuck behind one of the so called haystacks and either could not get back on the track or slipped about for some 20 to 30 seconds. I found the gear changing to be poor and at times non-responsive. The biker is more like a granny on a BMX.

S.D.E.



3D Lunattack

★ ★
Hewson Consultants Ltd
£7.95
CBM 64 + Joystick (optional)

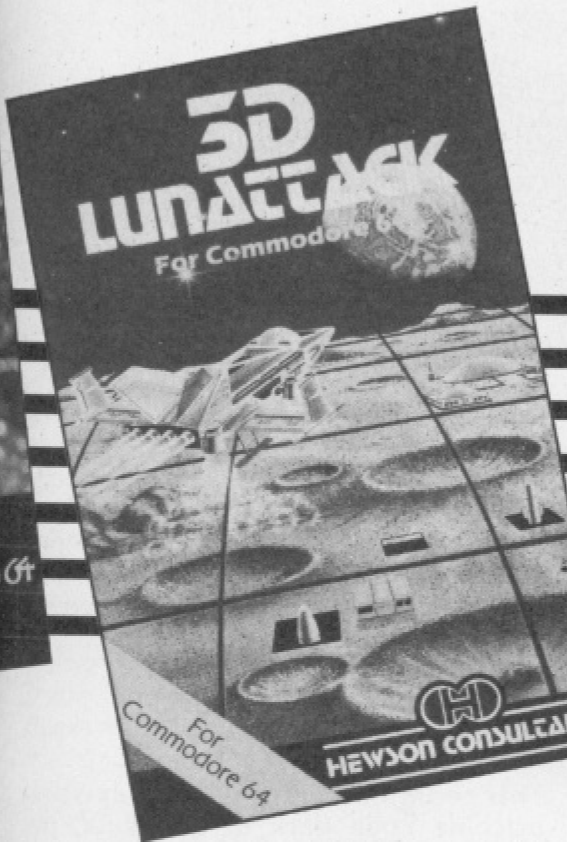
SET ON THE MOON, YOUR mission is to fly your Hoverfighter and destroy various Seiddab (baddies backwards) tanks, bases, missiles and so on. Your cockpit view shows a scrolling lunar landscape with instrument guidance for fuel, hull temperature and missiles. Navigation mode switches via the function keys to a map of your current location with the enemy forces displayed. Sounds good and obviously some thought has gone into constructing the game but sadly the package comes as a bit of a rough diamond – in need of a good cut and polish.

The screen intro did not impress. Details flashed by too quickly to read and I found myself constantly reaching for the cassette inlay. The user defined lettering was barely readable without considerable retuning. Once into the game shooting the enemy proved extremely difficult and whilst this game is supposed to be a tactical shoot 'em up rather than an all-out blastie, it left the impression that pixel point accuracy was needed to score a hit. Only the most dogged hoverpilot will be likely to better the resident high score of 10,000 points although it was good to see a two player and difficulty options.

Sound effects were reasonably good but the graphics of the lunar landscape were jerky and uninteresting compared with the cockpit. The navigation map was as murky as the lunarscape, somewhat reminiscent of a long forgotten Pink Floyd L.P.

It is a pity that the game falls flat in a number of minor ways but the inlay card assures us that the company's policy is one of constant improvement. Sticking to that policy could make Lunattack a good game.

R.M.





1985

★
Mastertronic/Severn Software
£1.99
CBM 64/ Joystick or keyboard

1985 IS A VARIATION ON THE old and rather tired 'Lunar Lander' theme. Your aim is to control a spacecraft as it visits four planets collecting Nuclear Pods, many of which are placed in very inaccessible positions. If you collect them all, you are allowed a stab at the final, most difficult cavern, where you will find the Fusion Core.

There is nothing new about the game, but it has been programmed in a sound, workmanlike manner. The



graphics are chunky but colourful, and the screen scrolls very smoothly. There is no background music, but the sound effects are spacey and effective.

The tape has a fast loader, and you are given the option of a lengthy demonstration sequence. It is from this demonstration that I gained most of my knowledge of the game, as I confess I found it incredibly difficult to play. The joystick control was unorthodox and I would defy anyone to manage using the keyboard!

I may be doing the programmer an injustice – I may be growing old and losing my grip, but I suspect that this game is for experts only, and for most of us is not worth buying.

P.R.B.

Kami-Kaze

★
Supersoft
£5.95(cassette); £7.95 (disc)
CBM 64 + joystick (optional)

THE STORY GOES THAT THE Xenon battlecruiser Argon, a spaceship by the way, landed on the pier at Cleethorpes fooling the locals into believing it to be a public loo and persuading them that the Xenon battle plans had the consistency of toilet paper. As all you chemistry buffs will know. Xenon and Argon are inert gases, i.e., not at all excitable.

This game, I'm afraid, is neither a gas nor an exciting one at that. The aim is to pilot your anti-matt saucer avoiding passing aircraft and clouds while annihilating the goodly citizens of Cleethorpes and their extremely well-trained dogs before they can take a call of nature and, you've guessed it, use up the loo-paper. The game has its humourous touches, though. The moggies tend to get tangled in the tissues as in the best T.V. ads but shooting our best friends when they've finished their business brings down the wrath of the R.S.P.C.A. in their helicopter.

The sound and graphics are both fairly simple but the splashes as the various corpses fall into the briny and the sprites used for the people and dogs are well done.

Use of the joystick, while optional, is preferable to keyboard operation especially if you hope to gain a free toilet roll at 10,000 points. There is a high score facility but no two player option. I was disinclined to beat my best far less attain the dizzy heights of free tissues and whilst the game may appeal to younger players, if you're looking for sophistication these days, forget toilet rolls.

R.M.

TEDMON

Brian Rushby's machine code monitor for the Commodore 16 should facilitate the production and de-bugging of machine code programs on Commodore's new machine.



THE COMMODORE 16 INCORPORATES a powerful single pass assembler/disassembler/machine code monitor, designed for the convenient production and de-bugging of programs written in 6502 (7501) machine code. The only problem is that, apart from a short description on how to enter the monitor (including a reference to a non-existent section) the manual does not say how to take advantage of this very useful facility. This article will explain the built in facilities.

TEDMON can be called either by typing MONITOR (or M shift 0) then RETURN or by pressing the RESET button whilst holding down the RUN/STOP key.

The monitor commands are listed and described below in alphabetical order. If at any time the computer locks just press the RESET button and you will recover with your machine code program still intact.

Entering commands

Commands can be entered immediately on entering the monitor. Any mistakes will be highlighted with a question mark. Standard convention is used in this article as described below.

$$$$ = hexadecimal address.
nnnn = hexadecimal number.
XXXX = decimal number.
< > = optional information.
< > = mandatory information.

The number of digits must be entered as stated (e.g. nn means two hex digits).

Command explanations

(1) Assemble

Assemble allows the entry of 6502 instructions which are directly translated into machine code. The monitor detects syntax errors and any branch out of range errors. Any mistakes will be highlighted with a ? and the instruction will not be translated. After successful entry of each line of code the monitor will prompt with the next available memory address on the next line (comparable with the AUTO line numbering in BASIC). All values must be preceded by a $ sign in the normal way for hexadecimal notation, also all numeric values must be preceded by a # .

Syntax: A $$$$(operand)

The operand is essential as outlined above < > (you must state which instruction to enter at memory location $$$$).

Example: A 2800 LDA # $00

This instruction will enter the start of your machine code program at memory location $2800 Hex (10240 decimal). The screen will then look like this:

```
A 2800 A9 00  LDA#$00
A 2802
```

The memory location $2802 is the next available; just type in the next operand and continue like this to the end of your program.

(2) Compare

This command allows the comparison of two sections of memory. The differences are reported in the form of memory location. When the two sections are identical nothing is reported.

Syntax: C $$$$ $$$$ $$$$
Example: C 0000 0100 0200

Where,

Address 1 = start of 1st section of memory. (e.g $0000)
Address 2 = end of 1st section of memory. (e.g $0100)
Address 3 = start of 2nd section of memory. (e.g $0200)

(3) Disassemble

This command translates existing machine code back to assembler, the reverse of assemble. Output will be ??? if the monitor encounters undefined bytes (e.g lists or tables). Corrections or amendments can be entered directly on top of mnemonics.

Syntax: D $$$$ ($$$$)

Example: D 0100

If this example, where only the first parameter is declared, a disassembly will be provided starting at memory location $0100 upwards for a screen full (21 memory locations).

Example: D 0100 0200

This example will give a full disassembly from memory locations $0100 to $0200.

(4) Fill

This command allows the programmer to set a complete block of memory to zero or any other value.

Syntax: F $$$$ $$$$ nn

Example: F 2800 2A00 00

This will set all memory locations between $2800 and $2A00 to zero.

(5) Go

Used to start execution of machine code program.

Syntax: G ($$$$)

Example: G

Start execution of m/c program from memory location already in program counter.

Example: G 2A00

Start execution at memory location $2A00

Your machine code program should always end with a BRK if passing control from the monitor. This ensures return to the monitor after execution.

(6) Hunt

This command will hunt through a specified block of memory for all occurrences of specified bytes, which can be declared as a string of hexadecimal numbers or a character string and can be up to 40 bytes long. Where a match is made the memory address will be displayed.

Syntax:- H $$$$ $$$$ nn (nn nn nn ....)
or H $$$$ $$$$ <string>

Example: H 0000 FFFF 42 41 53 49 43

A search between the memory locations 0000 and FFFF for the string of numbers shown. These are the ASCII code numbers that correspond to the word BASIC.

Example: H 0000 FFFF 'COMMODORE

A search is made for the character string COMMODORE

(7) Load

Used to load a program from tape or disc.

Syntax: L"Program name",X

Example: L"PROG",1

Loads a program from tape.

Example: L"PROG",8

Loads a program from disc.

(8) Memory

This command displays the HEX and ASCII codes of a specified block of memory. A reverse full stop is displayed where the HEX value does not have a corresponding ASCII character. Memory contents can be easily changed in this mode by entering the new values directly over existing values (use cursor keys to position over existing value).

Syntax: M $$$$ ($$$$)

Example: M 2800

This will give a display of 12 screen lines of memory starting at memory location $2800.

Example: M 2800 2A00

This will give a continuous display of memory from $2800 to $2A00.

(9) Registers

This command will display all the current contents of the CPU internal registers.

Syntax: R

Using this command will provide a display as below:

PC	SR	AC	XR	YR	SP
$$$$	nn	nn	nn	nn	nn

Where,

PC = Program counter.
SR = Status register.
AC = Accumulator.
XR = X Index register.
YR = Y Index register.
SP = Stack Pointer.

(10) Save

Allows your program to be saved to either tape or disc.

Syntax: S"PROG",x,$$$$,$$$$

Example: S"Prog",8,2800,2A00

This will save the program contained in memory block 2800 to 2A00 to disc under the name PROG. To save to tape substitute 1 for 8.

(11) Transfer

Used to transfer contents of memory from one block to another.

Syntax: T $$$$ $$$$ $$$$

Example: T 2800 2900 2A00

This will transfer the contents of memory locations 2800-2900 to the block of memory starting at 2A00 (and therefore ending at 2B00). All branch instructions will be automatically updated, however any JMP's or JSR's will need to be updated manually.

(12) eXit

Used to return control back to BASIC.

Syntax: X

Memory considerations

The Commodore 16 has 12K of free memory (RAM) for use by BASIC. This memory occupies addresses $1000 to $4000 (4096 to 16384 decimal). There is no free memory for use by machine code, we therefore need to release some of the BASIC memory. This is a simple operation, all we need to do is to limit the memory available to BASIC by POKing to the relevant registers.

First we need to decide how to split the 12K available between BASIC and machine code. If, for example, you wish to have 7K BASIC and therefore 5K m/c then the limit to BASIC would now need to be at memory address $2BFF (11263 decimal). The machine code can now be entered starting at memory location $2C00 (11264 decimal). We then need to POKE the new value to the limit to BASIC registers which are memory locations $0037 and $0038 (55 and 56 in decimal). These registers are in low byte-high byte order as is the normal convention.

The required values can be POKEd into the registers in immediate mode or in a BASIC program before loading in the m/c program, therefore it is best to work in decimal for this operation as BASIC does not recognise HEX. Calculate the values to POKE as follows:

Divide the limit to BASIC memory value by 256.

E.g $2BFF=11263
11263/256 = 43 (remainder=255)

Therefore 43 is the value to be poked into the high byte limit to BASIC memory register (56). The value to be POKEd into the low byte limit to BASIC memory register (55) is the remainder 255 as follows.

POKE 55,255:POKE56,43

Start entering your machine code at memory location $2C00.

TOP 20 Software

Compiled by Gallup

COMMODORE 64

TITLE		PUBLISHER
1	Ghostbusters	Activision
2	Daley Thompson's Decathlon	Ocean
3	Raid over Moscow	US Gold
4	The Staff of Karnath	Ultimate
5	International Football	Commodore
6	Hunchback 2	Ocean
7	Booty	Firebird
8	Zaxxon	US Gold
9	Bruce Lee	US Gold
10	Beach Head	US Gold
11	Combat Lynx	Durrell
12	Summer Games	Quicksilva
13	Football Manager	Addictive Games
14	Jet Set Willy	Software Projects
15	Select 1	Computer Records
16	Fighter Pilot	Digital Integration
17	Kong Strikes Back	Ocean
18	Blue Max	US Gold
19	Pyjamarama	Mikro-Gen
20	Impossible Mission	CBS

Retail sales for the month ended Jan 29 1985



VIC 20 Top Ten

TITLE		PUBLISHER
1	Perils of Willy	Software Projects
2	Hunchback	Ocean
3	Micky the Bricky	Firebird
4	Snooker	Visions
5	Jetpac	Ultimate
6	Vegas Jackpot	Mastertronic
7	Psycho Shopper	Mastertronic
8	Bullet	Mastertronic
9	Flight Path 737	Anirog
10	Duck Shoot	Mastertronic

Retail sales for the month ended Jan 29 1985

Compiled by Gallup for the industry's weekly trade magazine, Computer and Software Retailing. For details contact John Sorrenti, Computer and Software Retailing, 222 Regent Street, London W1R 3AB. 01-434 2131.

SETTING OUT ON AN ADVENTURE

Two of the problems covered this month in Allen Webb's series on creating your own adventure games are interpreters and data storage.

LAST MONTH I DISCUSSED THE DEVELOPMENT of a scenario and offered some general hints on how to create a convincing atmosphere. This month I want to get down to some more practical problems concerning the actual programming of the game. Specifically, I will deal with interpreters, more aspects of data storage and moving about.

The Interpreter

The key portion of any adventure is the interpreter. This section performs two functions. First it must split up the input command and extract the relevant words. This is called parsing. The early programs used simply two word, verb/noun commands such as TAKE WHIPPET or EAT PYTHON. Later programs extended the interpreter to accept full phrases. Probably the most sophisticated interpreter about is that used by Infocom. This allows complex commands including multiple sub-commands.

Once the command has been parsed, it is necessary to check the words to see if all are recognised. This can be a particularly slow process if written in BASIC. Listing 1 gives a combined machine code interpreter and word editor. Using the word editor you can enter the words you want the adventure to use and then save the interpreter plus word tables for use in your own adventures. You can have a maximum of 64 verbs, 128 nouns and 64 link words in the tables. That should be enough for most games.

This interpreter looks for three words in any command. Specifically, it looks for the first, second and last. These are called the verb, link word and noun respectively. The following examples will show what I mean.

Sentence	Verb	Link word	Noun
Go north	Go		North
Take the box	Take	the	box
Look in the large green box	Look	in	box

The interpreter exits with the position of each word in the relevant word table stored as follows:

Verb Position in 977
Link word Position in 979
Noun Position in 978

Program Listing 1

```
1 DATA76,65,194,162,0,169,32,157,141,3,157,162,3,157,183,3,232,224,21,208
2 DATA242,169,0,141,205,3,141,206,3,141,207,3,141,209,3,141,210,3,141,211
3 DATA3,141,213,3,169,47,141,215,3,169,13,141,218,3,169,51,141,216,3,169,2
4 DATA141,217,3,169,50,141,219,3,160,0,185,60,3,201,32,240,10,153,141,3,238
5 DATA205,3,200,76,71,192,172,205,3,204,204,3,208,3,76,239,192,200,162,0,185
6 DATA60,3,157,60,3,232,200,204,204,3,208,243,173,204,3,56,237,205,3,168,136
7 DATA169,32,153,60,3,200,204,204,3,208,245,173,204,3,56,237,205,3,141,204
8 DATA3,206,204,3,172,204,3,136,185,60,3,201,32,240,18,136,206,9,173,204,3
9 DATA141,206,3,76,177,192,238,206,3,76,152,192,173,204,3,56,237,206,3,168
10 DATA162,0,185,60,3,157,162,3,169,32,153,60,3,232,200,204,204,3,208,238
11 DATA173,204,3,56,237,206,3,240,25,141,204,3,206,204,3,160,0,185,60,3,201
12 DATA32,240,10,153,183,3,238,207,3,200,76,222,192,32,245,194,16?,0,133,187
13 DATA169,197,133,188,169,141,133,251,169,3,133,252,173,215,3,141,212,3,32
14 DATA167,193,173,208,3,201,255,208,6,32,222,193,76,153,193,141,209,3,173
15 DATA206,3,240,121,169,0,133,187,169,198,133,188,169,162,133,251,169,3,133
16 DATA252,173,216,3,141,212,3,32,167,193,173,208,3,201,255,208,37,169,0,133
17 DATA187,169,199,133,188,173,217,3,141,212,3,32,167,193,173,208,3,201,255
18 DATA240,7,24,109,219,3,76,101,193,32,222,193,76,153,193,141,210,3,173,207
19 DATA3,240,44,169,0,133,187,169,200,133,188,169,183,133,251,169,3,133,252
20 DATA173,218,3,141,212,3.32,167,193,173,208,3,201,255,208,4,32,222,193,96
21 DATA141,211,3,169,1,141,134,2,96,162,0,169,32,157,60,3,232,224,81,208,248
22 DATA96,169,1,141,208,3,160,0,177,187,209,251,208,6,200,192,4,208,245,96
23 DATA173,208,3,24,105,1,205,212,3,240,19,141.208,3,24,165,187,105,4,133
24 DATA1C7,165,188,105,0,133,188,76,172,193,169,255,141,208,3,96,169,1,141
25 DATA134,2,162,24,160,0,24,32,240,255,160,0,185,43,194,32,210,255,200,192
26 DATA21,208,245,169,34,32,210,255,169,5,141,134,2,160,0,177,251,20!,32,240
27 DATA7,32,210,255,200,76,4,!94,169,1,141,134,2,169,34,32,210,255,169,19
28 DATA32,210,255,169,255,141,213,3,169,1,141,134,2,96,73,32,68,79,78,39,84
29 DATA32,85,78,68,69,82,83,84,65,78,68,46,46,46,46,162,21,160,0,24,32,240
30 DATA255,169,5,32,210,255,160,0,169,32,32,210,255,200,192,40,208,248,162
31 DATA21,1C0,0,140,208,3,24,32,240,255,169,62,32,210,255,169,166,32,210,255
32 DATA32,154,193,32,228,255,201,0,240,249,201,32,208,8,174,208,3,240,240
33 DATA76,195,194,201,13,208,8,173,208,3,240,228,76,230,194,201,95,240,172
34 DATA201,42,240,89,201,20,208,29,172,208,3,240,208,206,208,3,32,210,255
35 DATA32,210,255,169,166,32,210,255,172,208,3,169,32,153,60,3,76,114,194
36 DATA201,65,48,179,201,91,16,175,172,208,3,192,38,240,168,153,60,3,238,208
37 DATA3,141,214,3,169,157,32,210,255,173,214,3,32,216.255,169,166,32,210
38 DATA255,76,114,194,173,208,3,141,204,3,32,2,195,76,3,192,76,239,192,162
39 DATA0,169,32,157,192,7,232,224,39,208,248,96,162,0,169,32,157,152,7,232
40 DATA224,39,208,248,169,1,141,134,2,169,18,32,210,255,162,23,160,0,24,32
41 DATA240,255,160,0,165,60,3,32,210,255,200,204,208,3,208,244,169,146,32
42 DATA210,255,169,19,32,210,255,169,6,141,134,2,96
43 DATA76,3,202,32,212,225,32,47,202,165,20,72,165,21,72,32,47,202,166,20
44 DATA164,21,104,133,21,104,133,20,173,232,3,240,4,169,54,133,1,169,20,32
45 DATA95,225,169,55,133,1,96,32,253,174,32,138,173,32,247,183,96
46 FOR I=49152TO49981
47 READ X: T=T+X: POKEI,X: NEXT
48 IF T<>107653 THEN PRINT"ERROR IN DATA"
```

This listing also appears in issue 7 (Feb/March 85) of 64 Tape Computing.

```
49 FORI=51712TO51768:READX:POKEI,X:NEXT:GOSUB50:GOTO51
50 FORI=50431TO51455:POKEI,32:NEXT:RETURN
51 V$="N   S   E   W   U   D   GO  T?KEGET EAT DRINLOOKEXAMOPENCLOSUNLOLOCK"
52 V$=V$+"KILLDIG DROPGIVE"
53 NV=64
54 N1$="NORTSOUTEASTWESTUP  DOWN"
55 N1$=N1$+"LAMPJUG KEY DOORSWORSHIEHELMSPADDAGGROPEDRAGGIANBREAWINEWELL"
56 N1=64
```

A zero value means that the command did not contain the type of word concerned, eg, if a single word command such as LOOK is given, then 978 and 979 will contain zero. If a word is not recognised, you will get the message. . . "I don't understand. . ." followed by the word concerned. Location 981 holds an error flag. If a command is rejected because a word is not recognised, this flag will be set to 255. The flag is set to zero if the command is accepted. Lines 176 to 189 show how to use the interpreter. When you call it, you will see a prompt near the foot of the screen. You can only input letters and RETURN enters the commands. DEL deletes the previous letter and " " erases the whole command. Pressing "★" will re-enter the previous command. So that you remember your last command, it is echoed in white below the input line.

The SAVE OBJECT CODE command will save the interpreter plus the word tables. These can be loaded in your own program by the line:

```
20 C=C+1:IF C=1 THEN LOAD "INTER-
PRETER",8,1
```

If you use this program, you should not use the area $C000 to $C800. This is deliberate to enable you to use the routines given last month with the interpreter in position.

To use the interpreter in your own games, the following two lines will be required:

```
100 SYS 12★4096: IF PEEK (981)=255
THEN 100
110 VE%=PEEK(977): PR%=PEEK(979):
NO%=PEEK(978)
```

I discussed last month the use of parallel execution to speed up the program. The relevant verb subroutine is selected by the simple line:

```
120 ON VE% GOSUB 1000,2000,...5000
1000 "LOOK"
2000 "TAKE"
5000 "DROP"
```

Data storage

Last month I touched on the storage of text behind the BASIC ROM. Clearly there is a large amount of other data which must be stored. Probably the key set of data used in an adventure is the location index. Using the examples given last month, this index is either the location number (small map) or the co-ordinate (large map). The following discussion is most relevant to the small map system.

On the whole, the use of variables or strings for the storage of data is inadvisable. The reasons are simple:

```
57 N2$="GOLD"
58 N2=64
59 P$="IN  UP  DOWNUNDEOUT THROINTOBEHITHE ON  OFF AT  "
60 NP=64
61 POKE49197,NV+1:POKE49207,N1+1:POKE49217,N1:POKE49212,N2+1:POKE49202,NP+1
62 FORI=1TOLEN(V$):POKE50431+I,ASC(MID$(V$,I,1)):NEXT
63 FORI=1TOLEN(N1$):POKE50687+I,ASC(MID$(N1$,I,1)):NEXT
64 FORI=1TOLEN(N2$):POKE50943+I,ASC(MID$(N2$,I,1)):NEXT
65 FORI=1TOLEN(P$):POKE51199+I,ASC(MID$(P$,I,1)):NEXT:PRINT"J"
66 POKE53281,0
67 POKE53281,0:PRINT"JXXX           #ADVENTURE INTERPRETOR"
68 PRINT"              ──────────────────"
69 PRINT"           #1. PRINT VERBS"
70 PRINT"           #2. PRINT NOUNS"
71 PRINT"           #3. PRINT LINK-WORDS"
72 PRINT"           #4. CHANGE A VERB"
73 PRINT"           5. CHANGE A NOUN"
74 PRINT"           6. CHANGE A LINK-WORD."
75 PRINT"           #T. TEST INTERPRETOR"
76 PRINT"           #S. SAVE OBJECT CODE"
77 PRINT"           <. SAVE TABLES"
78 PRINT"           >. LOAD TABLES"
79 PRINT"           \C. CLEAR TABLES"
80 PRINT"           #Q. QUIT"
81 GETI$:IFI$=""THEN81
82 IFI$="Q"THENEND
83 IFI$="T"THEN176
84 IFI$="S"THEN125
85 IFI$="C"THEN131
86 IFI$="<"THEN135
87 IFI$=">"THEN141
88 IFI$<"1"ORI$>"6"THEN81
89 SE=VAL(I$):POKE53281,4
90 ON SE GOTO 149,155,167,92,103,114
91 GOTO67
92 PRINT"J                #CHANGE A VERB"
93 INPUT"XXXXVERB NUMBER";VN
94 IFVN<1ORVN>64THEN92
95 PRINT"XXXINPUT #*# TO ERASE VERB IN TABLE"
96 INPUT"XXVERB";V$
97 LV=LEN(V$):IFLV=4ORV$="*"THEN99
98 FORI=1TO4-LV:V$=V$+CHR$(32):NEXT
99 V$=LEFT$(V$,4):IFV$="*"THENV$=CHR$(32)+CHR$(32)+CHR$(32)+CHR$(32)
100 TP=50431+(VN-1)*4
101 FORI=1TOLEN(V$):POKETP+I,ASC(MID$(V$,I,1)):NEXT
102 GOTO67
103 PRINT"J                #CHANGE A NOUN"
104 INPUT"XXXXNOUN NUMBER";VN
105 IFVN>128ORVN<!THEN103
106 PRINT"XXXINPUT #*# TO ERASE NOUN IN TABLE"
107 INPUT"XXNOUN";V$
108 LV=LEN(V$):IFLV=4ORV$="*"THEN110
109 FORI=1TO4-LV:V$=V$+CHR$(32):NEXT
110 V$=LEFT$(V$,4):IFV$="*"THENV$=CHR$(32)+CHR$(32)+CHR$(32)+CHR$(32)
111 TP=50687+(VN-1)*4:IFVN>64THENTP=50943+(VN-65)*4
112 FO?I=1TOLEN(V$):POKETP+I,ASC(MID$(V$,I,1)):NEXT
113 GOTO67
114 PRINT"J              #CHANGE A LINK-WORD"
115 INPUT"XXXXLINK-WORD NUMBER";VN
116 IFVN<1ORVN>64THEN114
117 PRINT"XXXINPUT #*# TO ERASE LINK-WORD IN TABLE"
118 INPUT"XXLINK-WORD";V$
119 LV=LEN(V$):IFLV=4ORV$="*"THEN121
120 FORI=1TO4-LV:V$=V$+CHR$(32):NEXT
121 V$=LEFT$(V$,4):IFV$="*"THENV$=CHR$(32)+CHR$(32)+CHR$(32)+CHR$(32)
122 TP=51199+(VN-1)*4
123 FORI=1TOLEN(V$):POKETP+I,ASC(MID$(V$,I,1)):NEXT
124 GOTO67
125 PRINT"J                \SAVE OBJECT CODE"
126 INPUT"XXDEVICE NUMBER";DN
127 INPUT"XXFILE NAME";F$
128 SYS 51712 F$,DN,2,49152,51455
129 IFDN=8THENGOSUB191
130 GOTO67
131 PRINT"J                \CLEAR TABLES"
132 INPUT"XXARE YOU SURE";IN$
133 IFIN$<>"Y"THEN67
134 GOSUB50:GOTO67
135 PRINT"J                \SAVE WORD TABLES"
```

Program Listing 1

```
136 INPUT"■■DEVICE NUMBER";DN
137 INPUT"■■FILE NAME";F$
138 SYS 51712 F$,DN,2,50431,51455
139 IFDN=8THENGOSUB191
140 GOTO67
141 PRINT"■                ▶LOAD WORD TABLES"
142 INPUT"■■DEVICE NUMBER";DN
143 INPUT"■■FILE NAME";F$
144 FORI=1TOLEN(F$):POKE678+I,ASC(MID$(F$,I,1)):NEXT:POKE183,LEN(F$)
145 POKE184,1:POKE185,0:POKE186,DN:POKE187,167:POKE188,2:POKE780,0
146 POKE781,50431-INT(50431/256)*256:POKE782,50431/256:SYS65493
147 IFDN=8THENGOSUB191
148 GOTO67
149 PRINT"■            VERB TABLE":PRINT
150 TP=50432:FORI=1TO64
151 PA$="..■":IFI<10THENPA$="...■"
152 PRINT"■"I;PA$;:FORJ=1TO4:CH=PEEK(TP):IFCH=32THENCH=45
153 PRINTCHR$(CH);:TP=TP+1:NEXTJ,I
154 GOTO173
155 PRINT"■            NOUN TABLE":PRINT
156 TP=50688:FORI=1TO64
157 PA$="..■":IFI<10THENPA$="...■"
158 PRINT"■"I;PA$;:FORJ=1TO4:CH=PEEK(TP):IFCH=32THENCH=45
159 PRINTCHR$(CH);:TP=TP+1:NEXTJ,I
160 PRINT"■■■          ■■PRESS SPACE FOR MORE"
161 GETI$:IFI$<>" "THEN161
162 PRINT"■            NOUN TABLE":PRINT
163 TP=50944:FORI=1TO64
164 PA$="..■":IFI>35THENPA$=".■"
165 PRINT"■"I+64;PA$;:FORJ=1TO4:CH=PEEK(TP):IFCH=32THENCH=45
166 PRINTCHR$(CH);:TP=TP+1:NEXTJ,I:GOTO173
167 PRINT"■          LINK-WORD  TABLE":PRINT
168 TP=51200:FORI=1TO64
169 PA$="..■":IFI<10THENPA$="...■"
170 PRINT"■"I;PA$;:FORJ=1TO4:CH=PEEK(TP):IFCH=32THENCH=45
171 PRINTCHR$(CH);:TP=TP+1:NEXTJ,I
172 GOTO173
173 PRINT"■■■          ■■PRESS SPACE FOR MENU"
174 GETI$:IFI$<>" "THEN174
175 GOTO 67
176 PRINT"■"
177 PRINT"■■VERB NUMBER     ■■■"   "
178 PRINT"■LINK-WORD NUMBER     ■■■"
179 PRINT"■NOUN NUMBER     ■■■"
180 PRINT"■■■■■■■PRESS ■← TO EXIT"
181 PRINT"PRESS ANY OTHER KEY TO CONTINUE TEST"
182 SYS12*4096:IFPEEK(981)=255THEN182
183 PRINT"■■VERB NUMBER     ■■■"PEEK(977)
184 PRINT"■PREP NUMBER     ■■■"PEEK(979)
185 PRINT"■NOUN NUMBER     ■■■"PEEK(978)
186 PRINT"■■■■■■■■PRESS ■← TO EXIT"
187 PRINT"PRESS ANY OTHER KEY TO CONTINUE TEST"
188 GETI$:IFI$=""THEN188
189 IFI$<>"←"THEN176
190 GOTO67
191 OPEN15,8,15:INPUT#15,EN,EM$,ET,ES:CLOSE15
192 PRINT"■■DISK STATUS.."EN;EM$;ET;ES
193 FORI=1TO3000:NEXT:RETURN
```

i) The use and manipulation of strings can involve garbage collection and its attendent delays.
ii) An integer variable uses 2 bytes and a floating point variable takes 5 bytes.
iii) Variables steal your space for BASIC.

A better approach is to store data as simple arrays of single or double bytes POKEd into some safe area of RAM (eg behind the KERNAL using the routines given last month). Each value can then be accessed simply by PEEKing the value determined by an offset. This approach is the machine code type approach and may make life simpler if you choose to move from BASIC to assembler. So what data must we have? The normal information is:

i) Location of objects/people
ii) Movement information
iii) Flags indicating the status of doors etc.

It is possible to envisage three sets of conditions for any objects in an adventure. Provided you don't have more

Program Listing 2

```
400 IF VE%<9 AND VE%>0 THEN DI%=VE%: GOTO500
410 IF NO%<9 AND NO%>0 THEN DI%=NO%: GOTO500
420 :
430 :
440 :
450 :
500 IF PEEK(SA+LOC) AND DI% >1 THEN 520
510 PRINT"YOU CANNOT GOT THAT WAY": RETURN
520 LOC=PEEK(LOCTAB+(LOC-1)*10+DI)
530 REM DESCRIBE NEW LOCATION AND RETURN
700 :
710 :
720 :
730 :
1000 PRINT"I CAN SEE:-": FL=0
1010 FOR ON = 1 TO N :REM CHECK ALL OBJECTS
1020 IF PEEK(OT-1+ON)<>CL THEN 1040
1030 PRINT OBJ$(ON): FL=1
1040 NEXT ON
1050 IF FL=0 THEN PRINT"NOTHING USEFUL"
1060 :
1070 :
1080 :
1090 :
1100 :
2000 IF PEEK(OT-1+NO%)=CL THEN 2020
2010 PRINT "IT'S NOT HERE":RETURN
2020 POKE(OT-1+ON),255: PRINT"TAKEN":RETURN
```

than 254 locations, this data can be represented by a single byte:

i) Object non-existent...byte value is 0
ii) Object in your possession...byte value 255
iii) Object in location...byte value is location number.

Any commands such as TAKE, LOOK, INVENTORY, EXAMINE etc. need only refer to and update this table. Imagine that you have ON objects, each with a description OBJ$ in a game. The position of the objects is in a table starting at address OT and CL is the current location. The routine between lines 1000 to 1050 in listing 2 is an example LOOK routine, and that between lines 2000-2050 performs TAKE.

The storage of movement data is a little more complex. In effect we want to have access to two sets of data:

i) Which routes are valid for any given location.
ii) Where does that route take you.

The usual directions used are the four cardinal points of the compass, the four half cardinal points and up and down, in all ten directions. The number of locations with up or down will normally be small and can usually be dealt with in the main program logic. That leaves eight directions which can be represented by one byte per location. Let us consider each bit of a byte to be a direction flag . . . set means you can go that way, zero means you can't. Additionally let bit 1 represent northeast etc. The following examples will show my meaning:

The next step is to ensure that the eight directions occupy the same positions in the verb and noun tables, so that north is 1, northeast 2 etc. The routine from lines 400 to 530 in listing 2 shows how to test valid moves. Line 400 traps commands such as NORTH, WEST etc. Line 410 traps commands such as GO NORTH, MOVE WEST etc. Line 500 checks if the route is open or closed. The routine assumes that the location number is in LOC and the direction table starts at address SA.

Line 520 in the listing updates the array holding the movement data (starts at address LOCTAB). Each location entry has ten bytes, one for each direction. Each byte contains the number of the location reached if you move in that direction. For example, if going east from location 5 takes you to location 10, then the third byte of the entry for location 5 will contain 10.

Directions	Bit Pattern	Byte Value
N	00000001	1
all directions	11111111	255
E	00000100	4
SE,NW	10001000	136



Next month, I'll talk about data compression and artificial intelligence.

BUSINESS FILE

Take the strain out of writing letters to your bank manager or typewriting your latest novel by transforming your Commodore into a word processor with some help from Dave Crisp.

IN THE LAST TWELVE MONTHS OR SO printers have gone from an expensive luxury to a reasonably priced necessity. It is now possible to get a very good dot-matrix printer or one of the lower priced daisy-wheel printers for less than £200 with careful shopping around. Because of this reduced cost, word processing software is beginning to catch on. The quality of word processing software has also improved dramatically and it is now quite reasonable for a business to use a home computer as a dedicated word processor.

Disadvantage

The selection of word processors currently available range in price from £14.95 to around £100. The only problem with almost all these, which is due to the machine rather than the software, is that you have to do your word processing on a 40 rather than an 80 column screen. Initially, I continued to use my 8096 which has an 80 column screen with the text editor which is provided with the Silicon Office. That was until I found a word processor which is so good that I felt it's worth putting up with a 40 column display.

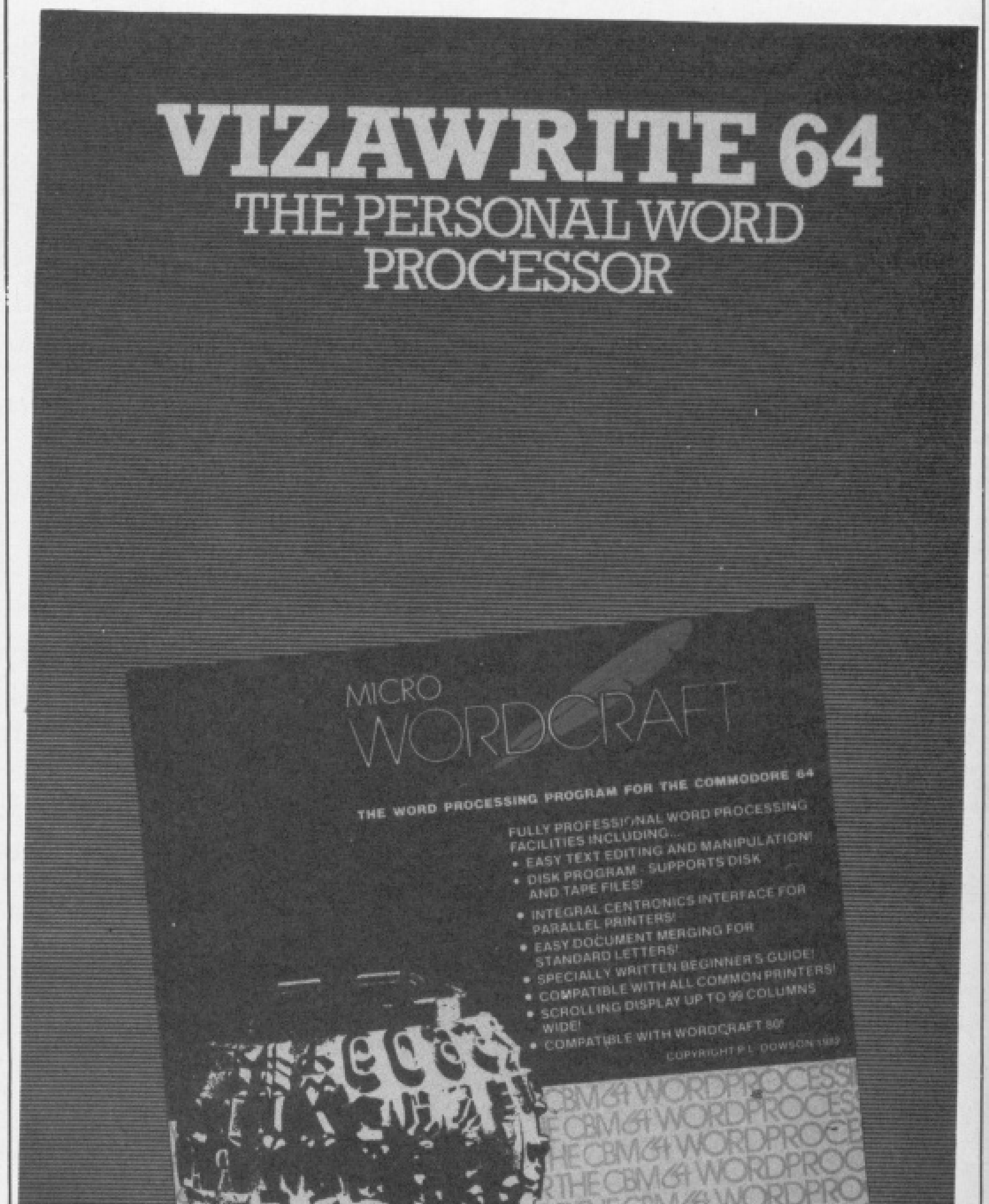
One man's meat is another man's poison and this is particularly true with word processors. Some people like the way a wordprocessor manipulates text and the ease with which it can be done, other people go for features. It really depends on you and your particular requirements.

Please bear in mind that my use of a word processor is mainly for typing out large pieces of text for articles and so I am inclined to go for something that allows me to work fast. Your needs may differ.

Micro Wordcraft

Micro Wordcraft from Dataview is the baby brother of a successful word processor on the large Commodores.

The manual goes from square one and presumes you know nothing. After loading the program; a start-up menu appears. This allows you to set up the particular disc/tape/printer configura-



tion that are using. Having made your choice you move onto the text editing screen.

My first impression of the page was that it was messy: the line length and column number, for example, squeezed in. The page width defaults to 75 char/line and you are almost ready to type.

In order to enter commands e.g. centre text, move text and so on, you press the run/stop key. This moves the cursor up to the top line so that you can enter the command you require. Pressing it again puts you back into edit mode. Apart from the logical use of the run/stop key, I had a job remembering which key did which. For instance, to jump around the screen you had to press the Commodore key and P. I would have found the J easier to remember. To set a tab, you using the Commodore and 3 key. This type of key use does not produce fast text entry.

Printing out

There is no doubt that Micro Wordcraft is set up to print with virtually any printer type you have but setting up page widths, line spacing, control characters could have been easier.

It is possible to set up headers and footers which are printed at the top and bottom of each page.

Mail merge

Wordcraft call their mail merge facility 'fill'. It is fairly easy to use. As with all mail merge facilities, it enables the user to merge a file containing text – say, a letter – with another containing a list of items – say, names and addresses – for mailshot purposes, for example.

Sounds bad

Wordcraft really isn't as bad as it seems. It takes practice. My two main complaints are the way the page scrolls when typing in (hard to describe) and the way characters are left on the screen to indicate which parts are to be underlined, etc..

Still, it is not too bad and I would be quite happy to use it if I could not use my favourite.

Easyscript

I will get hung drawn and quartered for this but this is possibly the worst word processor I have ever used! Its one saving grace is that you can switch out of editing mode to view your page. This means scrolling over the page but at least you get a good idea of how things will look printed out.

The manual is as exciting as watching milk curdle. It leaps about from one thing to the next and leaves you cold. All the important things are there – word wrap, justification, commands to underline, embolden etc but when you use them, you can find yourself with what appears to be a list of unconnected words surrounded by hieroglyphics.

Printing out

There is a built-in software interface for parallel printers which works with most leads, and enables you to get at most of the features of an Epson FX80, for example. A little experimentation is required in order to obtain italics etc but, if you dig into the manual, all the information is there. RS232 users are catered for as well and so Easyscript does score in the printout department.

Retrospect

Again, I have appeared to concentrate on what I do not like as opposed to what I do like. There are features which make this fairly good although not for my own use.

The mail merge is there and is comprehensive. Find words, replace words, etc are all there. Nothing drastic is missing. Before you buy a word processor try and find somebody with Easyscript, it should not be too hard, and have a look. You may like what you see.

Vizawrite

This is the one – the best, the easiest, and the cleanest. It was so good it made an 80 column screen unimportant. It does all the things I require of a word processor and it does them in exactly the way I would choose to do them if I was designing a word processor for my own use. The only thing I would really love now is the cartridge version.

As far as features go it is much the same as the other two. Everything is so easy you barely need a manual. If you want to delete a block of text you press the Commodore key and delete, then cursor over what is to go. If you want to insert text you press the Commodore key and insert. If you want to move to a tab region you press the CTRL key and tab. The function keys have been used to the full and with Vizawrite I can blast away and type text as I think about it without having to worry about formatting.

See all as you type

You can set the line length where you want it and type so the screen scrolls with the cursor or you can change to width mode. This compresses your text into 40 column mode and means you see all you type. With two key presses you can revert to how it will print and be sure all still looks good.

No squiggles

The nice thing with this is that if you input a format command, you don't get squiggles and lines indicating it is there. Most of the marking is done with text inversion or colour change. Some care has to be taken with your choice of screen colours as some marking may not show. For instance, if you want to delete text, the text you move over is highlighted in white. If you have chosen white text then, of course, this will not show up. On a green screen monitor text colour is even more important but, if you take the advice in the manual, all will be OK.

Hardcopy

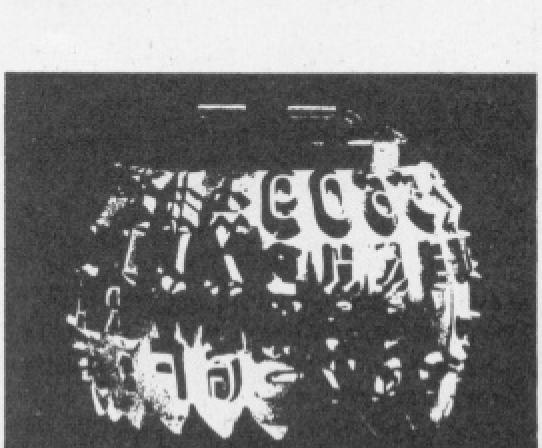
Printing out is easy. Pressing the Commodore key and P takes you to the printout menu. Here you can toggle on and off things such as line feed, justification, and so on. You can specify the column to start printing all the important little things. You can set up the word processor to send specific escape codes to get at particular features of your printer and the method of doing this has been improved on later versions.

Favourite

This is my personal favourite – I am using it to type this article. It is certainly well worth looking at.

Wordprocessor features

To conclude, the following are some features which I think all word processors should include: find and replace a word, block move, block delete, headers, footers, word wrap, justify, page preview, wide range of print options, mailmerge/fill, underline, embolden (printer dependent) ease of use at speed, save paragraph, load paragraph, set page colours. I hope this helps you in choosing a word processor to suit your needs.



Your COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

SO YOU OWN A COMMODORE?

SO YOU'VE WRITTEN SOME PROGRAMS?

SO WHY HAVEN'T YOU SUBMITTED THEM TO US?

Your Commodore is always on the lookout for new material for publication and we know that there are thousands of intelligent, literate, innovative and creative Commodore owners out there, so why don't we get together?

If you have written an exhilirating game or an invaluable utility on your Commodore micro, share your talents with us and our readers by submitting your efforts and the form to the address below. All articles should be documented and type-written and should be accompanied by a printout of the program as well as a copy of the program on cassette or disc. All material should be original; if it is not chosen for publication, it will be returned to you.

You may not have written any software yourself, but you have very firm opinions about the world of Commodore and all their attendant industries and products. Then put your opinions on paper and post them to us, again at the address below — you never know, you might even get paid for airing your views! All submissions should be sent to:

The Editor
Your Commodore
Argus Specialist Publications Limited
No 1 Golden Square
London WIR 3AB

★**PLEASE COMPLETE IN BLOCK CAPITALS**

Your Name

Program Name

Computer/memory size it runs on

Amount of memory program occupies

Other computers/memory size which your program runs on without conversion or use

Does your game need or use joysticks? Yes No

Have you sent your game to another magazine Yes No

Is it original/or a variation on a theme?

Your Address

Telephone Number

Times to contact you


A Tony Crowther/Quicksilva Production
BLACK THUNDER
Coming to your screen soon!
COMMODORE 64
Cassette £7·95 Disk £12·95
Available exclusively from
QUICKSILVA
DJR
in association with The Wizard Development Company Ltd.
Argus Press Software Group
Quicksilva Mail Order, P.O. Box 6, Wimborne, Dorset BA21 7PY.
Telephone (0202) 891744.
Selected titles available at:
WHSMITH
Boots
WOOLWORTH
John Menzies
and leading multiple retailers and specialist computer stores.
GSH

BOARD

Phil South has been scrabbling around the computer versions of famous board games. Find out if he was bored.

CHRISTMAS AT MY FAMILY'S ALWAYS involved, at some point, the ritual of a full scale game of Monopoly. I always found this fascinating, as it had the tang of the 1950's about it, and as a consequence I will associate board games with Christmas forever. Now, the other day, I was shocked by the information that Leisure Genius had conspired with Waddingtons, and other leading board game manufacturers, to produce the definitive computer versions of all my childhood favourites. Don't get me wrong, I love computers and I'm a confirmed computer game addict, but I'm a bit fastidious where board games are concerned, I hold the belief that the fun of a board game is mainly in the actual mechanics of play, rolling dice etc.

How can these automated versions match up to the original Monopoly, Kensington, Scrabble, Cluedo, and Mastermind? Can they play a decent game? What's the point? With an open mind and a spare afternoon I decided to find out.

Do not pass go, do not collect £200

Monopoly is a version of the world famous best selling property game. (I'm saying this for the benefit of anyone who's been on Alpha Centauri or living with apes in Africa for the last fifty years!) The point of it is to accumulate enormous amounts of money, and be a cut-throat business tycoon, collecting rent from the other players and sending their little businesses to the wall (a real game of the eighties, this one). You do this by moving your pieces around the board according to the throw of two dice and deciding whether or not to buy the properties you land on. If you don't buy the property, it is auctioned off to the highest bidder. When you buy a road you can build houses or hotels and charge rent to anyone who lands on it.

Community Chest and Chance cards give you random directions to rock the boat a little, just to even things up for the less canny players. (Surely there can't be anyone who hasn't played this game? By the way, how are things on Alpha Centauri?) The game is built on shrewd management and trading, and people with a mercenary streak will thrive.

When you boot up Computer Monopoly, a lovely 3D rendering of the famous board appears, as if looking at it from your position at the table. Beneath this view is displayed the board squares in the vicinity of your piece, the square you are on being on the extreme right. You choose the piece you want to play, and as you press the button to roll the dice they are "thrown" in from outside the screen, to land squarely in the centre of the board. This game is full of nice animation touches, like the way the game pieces move around the board. With thoughtful programming like this the flavour of the original game is held, but the game moves much faster. You can play short games, timed games, and indeed full-length family Christmas-type games, too! For insomniacs, there is also the facility to set up the computer to play you.

Of the bunch, this is the best conversion of the lot, and if you like Monopoly you'll enjoy this package enormously.

Whodunnit?

Cluedo is murder. No, I mean, Cluedo is a game of who killed who with what and where, based on Ludo. Aspiring sleuths will know this one already, but in case you never saw it, it is a version of the famous detective game, where you must solve, with a combination of investigation, skill and dumb luck, who murdered Dr. Black. The Doctor was found at the bottom of the stairs at his home in Tudor Close, and to win the game you must, in true whodunnit dramatic style, accuse the murderer, specify the weapon used, and say where the crime took place.

When the crime takes place, the police arrive (nick-nick) and clap eyes on six suspects, Col. Mustard, Prof. Plum, Rev. Green, Mrs. Peacock, Miss Scarlett, and Mrs. White. There is also a gruesome selection of weapons, a dagger, a candlestick, a pistol, a piece of rope, a spanner, and of course, the obligatory blunt instrument, a piece of lead pipe. Computer Cluedo can be played by 2-6 players, or you against 1-5 computerised opponents. The board is on screen, and as you enter rooms you get a lovely 3D rendering of the room from the doorway. As each player takes his or her turn, their "signature tune" is played; these are appropriate ditties, which make good use of the excellent SID synthesiser chip.

I like this, and while the original game wasn't as brilliant as, say, Monopoly, the computer version is lively, colourful and fun to play with.

Red, red, white, blue ...wrong!!!

Mastermind was originally a simple logic

game; one person set up a hidden code made up of four coloured pegs from a selection of six colours, and the other person had to guess the code in as few guesses as possible. As play progressed, the codemaker gave the codebreaker cryptic clues in the shape of black and white pegs, indicating colour and position correct or colour only correct respectively. Latterly, things got a trifle out of hand, with the introduction of new versions of the game including numbers, shapes, or in the worst cases, all three! Computer Mastermind allows you to play any of these versions of the game, against your chums or against the computer. As with the other games, an on screen board is generated, as are appropriate (?) noises during the game.

I got a version of the original Invicta Plastics game when it came out, and I loved it. This computerised version is a bit lifeless, but it plays a good game of Mastermind, and is good value from the point of view of having every different type of the game included in your options. Adequate but not stunning.

There's no such word as floonk

Scrabble is a word game in which you produce interlocking words (crossword style) using little plastic lozenges, called "tiles", with letters embossed on them. Each letter has a different value or score, which is inversely proportional to its difficulty to place in words from the English language; letters like "z" or "x" have very high scores. All players are given seven tiles and they must create proper English words, replacing letters as they are used so that they always have seven tiles.

The winner is the person who exceeds 500 or 700 points, depending on the skill level selected. Scores are collected by the sum of a word's letter scores, and multiplied by the factors indicated on some of the squares on the board; triple word score, triple letter score and so forth. There are other rules, such as anyone who ends the game with some tiles left has those scores deducted from his total, and if some smart alec uses all his tiles with the last word his score is increased by the sum of all the other players remaining tiles, lucky so-and-so! Computer Scrabble is totally faithful to the original game, with of course the option to play the computer if you feel like a game and can't find a sucker to play you. It also features an on screen board, and offers the benefit of having the computer doing the adding up for you, if you, like me, are totally innumerate.

Scrabble is a classy game, a sort of a thinking man's Beano Word Puzzle, and in this computer-aided incarnation loses none of the original appeal.

Phase one, place your stones

Kensington started as a high class option to having a chess board on your coffee table, with an expensive advertising campaign carrying endorsements from Omar Sharif and other celebrities with a reputation for intelligence. It is a game of strategy, logic and skill on a board made up of hexagons, squares and triangles. Like most games of strategy, the aim is to gain territory within the restricting you-go-I-go structure of a game play, trying to conceal your intentions when your opponent has a clear view of your pieces. The aim of this particular game is to "capture" the hexagons, either the white ones or your own colour, by surrounding them with your "stones". The computer version, again, has the option of playing against real players or the computer. The placement of stones is achieved by using the keys WER, SD, and ZXC as a kind of "joystick" (look on your computer keyboard and you'll see why), shuffling the stones laboriously from the preset start position, to where you wish to drop them.

This is really the worst of the lot. The placement of the stones is so tedious; there are more than eight angles for the stones to travel along, and only eight keys. It falls to a matter of guessing which key will take the stone in the direction you want it, and if you guess wrong a horn sounds and the words 'wrong direction' flash on the screen for what seems like for ever, and the keyboard siezes up for a few seconds. I wouldn't mind, but it's not even part of the game; in the real game you would just put it on the spot you wanted. This computerised process wastes time and causes enormous frustration, and to top all this, it doesn't even play a very good game of Kensington.

And the winner is . . .

So, what does all of this tell us? What about fact that the computer can, in all these games, be programmed to play itself and why is more time spent on the graphics than on improving the playability? I think my final word is this: computerising board games, if intelligently done, is fun and worthwhile, like Monopoly, but bad programming and just plain woolly thinking turns them into useless monstrosities. Full marks for your coding Leisure Genius, but minus several points because on the strength of these games I can see there is a lot of original programming talent going to waste!

Leisure Genius are at 3 Montague Road, London. (01-935 4622). All the games cost £12.95 each.

S 1

T 1

I 1

F 4

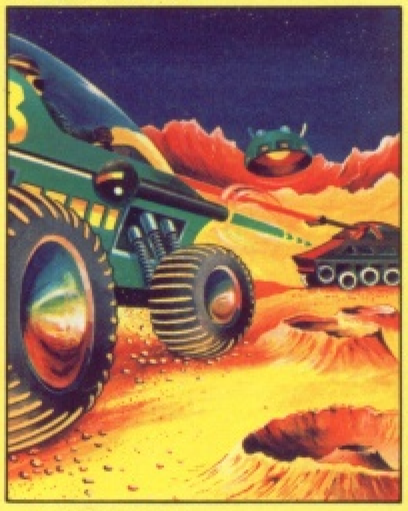
F 4

COMMODORE C16 AND PLUS 4

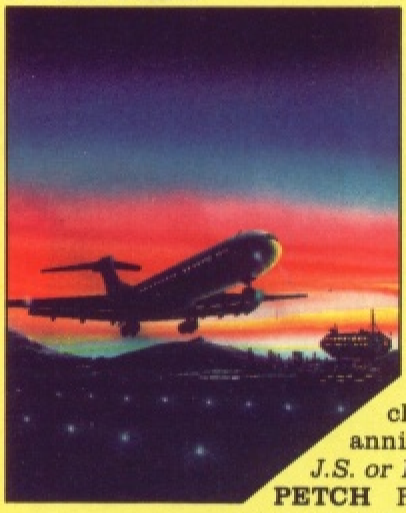
LAS VEGAS

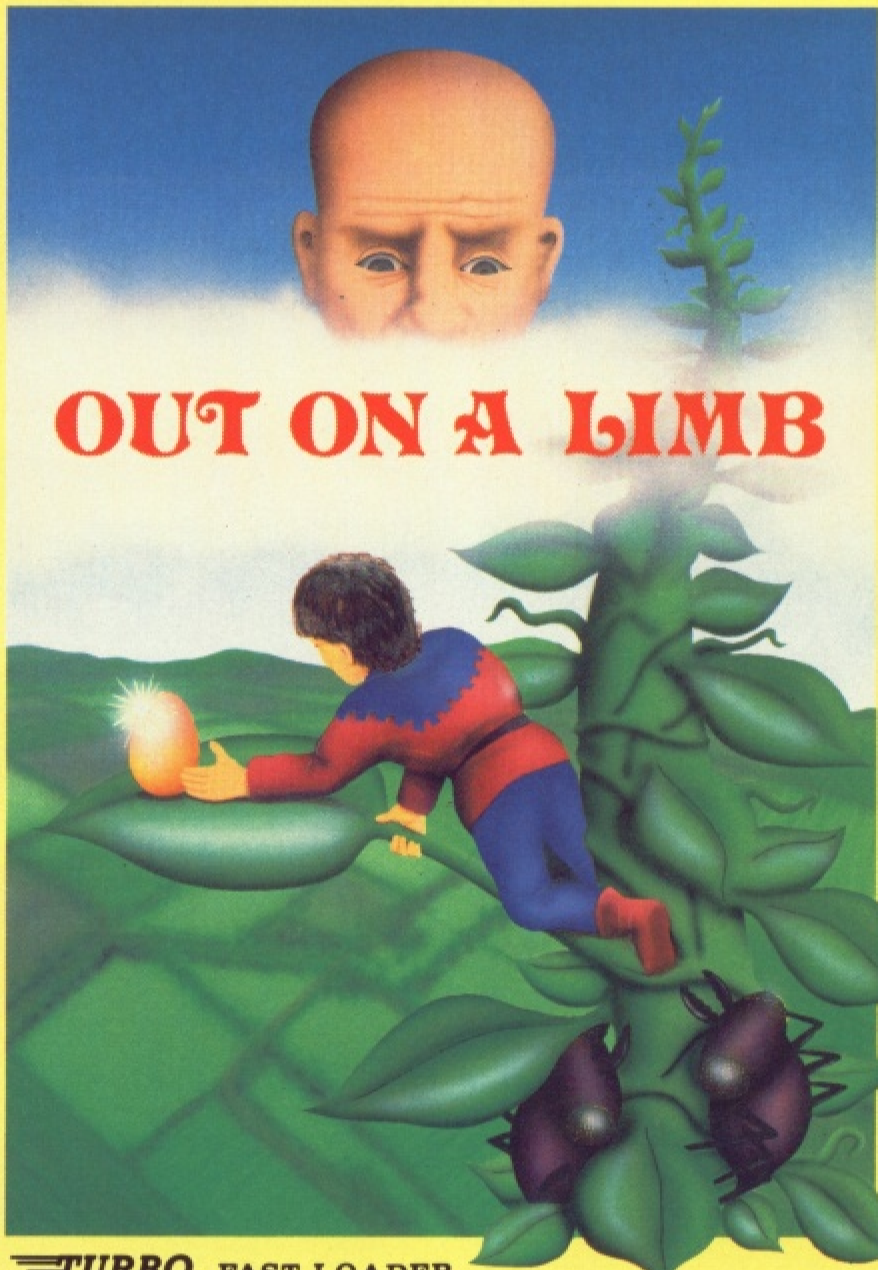


MOON BUGGY



FLIGHT PATH





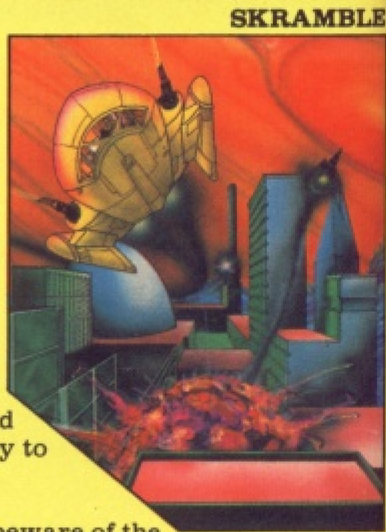
ZODIAC



3D TIME TREK



SKRAMBLE



TURBO **FAST LOADER**

LAS VEGAS Las Vegas brings all the challenges of a deluxe arcade fruit machine direct to your fingertips. Many exciting features include, gamble, cancel, collect, nudge and number options, with a three row display and maximum payout of 20$.
K.B. Only C.16 £6.95

ZODIAC This is an enthralling arcade adventure in which the evil powers of hell have scattered the signs of the Zodiac in the four hundred chambers of the abyss. Struggle to collect these signs and at the same time try to annihilate everything in your way. How long can you stay alive?
J.S. or K.B. C.16 £6.95

PETCH Race Petch around the screen, moving ice blocks to alter the maze, however, beware of the nasty monsters who are constantly chasing Petch as you attempt to collect the bonus cherries. Also if you can touch the edge of the maze the monsters will suddenly burst into flames, but look out they'll soon be back!
J.S. or K.B. C.16 £6.95

MOON BUGGY You must skillfully manoeuvre your jumping patrol vehicle over dangerous moon craters as well as large boulders and cunningly placed mines. Not only this but avoid the hovering alien spaceship as it bombards you from above.
J.S. or K.B. C.16 £6.95

3D TIME TREK As sole survivor of the planet "Corillian" your quest is one of anger and revenge. The starship you are flying is full of the latest inboard computers and extra powerful sensors. Also included are full 3D graphics, to add unbelievable realism to this fantastic journey through time itself, and beyond.
J.S. and K.B. C.16 £6.95

SKRAMBLE Earth has been overrun by the Cobrons and its up to you to battle through the six ferocious and testing sectors. Adversaries include meteors, UFO's and deadly fireballs. Fly through an armoured city, then an elaborate maze and finally the command base itself.
J.S. or K.B. C.16 £6.95

FLIGHT PATH Flight Path is without doubt the best flight simulator on the C/16 and Amstrad. The many elaborate features include; Altometer, flaps, directional headings, crosswinds, fires, ground warning lights and reverse thrust to name but a few. Also included are smooth graphics as you take off, cruise over mountains, and land once again.
J.S. and K.B. C.16 £6.95

OUT ON A LIMB This is a fantastic and in parts outrageously funny game. Based on the fairytale of Jack and the Beanstalk, Out On A Limb is full of the most strange and eccentric characters you are ever likely to meet. Firstly, climb the stalk and jump onto the clouds, then enter the giant's castle searching for treasures. However, watch out for vacuum cleaners, musical notes, televisions and potted plants, all of which chase you round the many and elaborate rooms of the castle. Once the treasure is collected the single exit will be opened, and then...........?
J.S. or K.B. C.16 £6.95

ANIROG

Mail Order: 8 HIGH STREET HORLEY SURREY 24 HOUR CREDIT SALES HORLEY 02934 6083 Payment by: P.O. - ACCESS - VISA

AVAILABLE FROM YOUR COMPUTER STORE

AD INDEX

Mastering Machine Code

Although many programmers may disagree, a grounding in BASIC can be useful to the machine code programmer. A P and D J Stephenson show you how.

IN PREVIOUS PARTS OF THIS SERIES, WE HAVE concentrated on the instructions and addressing modes peculiar to the 6510A. We shall assume, from now on, that these are understood or at least you know where to look them up. There are still a few more to come but it is time now for consolidation.

Machine code via BASIC

Apart from masochists and members of MENSA, most of us who buy home computers start off with BASIC. It is an easy language to learn and only a few weeks pass before the average beginner is able to cope with most of the common programming tricks. The question is, whether or not a preliminary apprenticeship in BASIC prepares you for tackling machine code later on? Some purists say no – in fact one or two of them preach that previous familiarity with BASIC can permanently damage the intellect! Still, purists are good things to have around providing we don't always take them as seriously as they take themselves. We must face the fact that, in the real world (the microcomputing part of it anyway), the newcomer to machine code will still tend to 'think' in BASIC. In other words, he/she will be constantly trying to find some relation between a BASIC routine and a corresponding machine code equivalent. It sweetens the pill a little if a gentle transition to machine code is made via a series of BASIC equivalents. In those which follow, we should point out that:

(a) There may be ways, other than those given, for arranging the equivalent machine code.
(b) Symbolic operands and labels are used. Suitable absolute addresses must be chosen, or assigned by an assembler if you want to try them out.
(c) Single byte numbers are assumed. That is to say, numbers must be kept within the bounds of +127 or –128 if you use two's complement working or, in unsigned binary, 255. Remember that numbers may start within the limits but after a few machine code steps, the limit may be exceeded.
(d) If more than one machine code equivalent is given, they will be numbered 1, 2, etc.

BASIC/Machine code equivalents

Initial assignment of constant:

```
BASIC               Machine code
VELOCITY=65         1. LDA # 65
                       STA VELOCITY
                    2. LDA # $41
                       STA VELOCITY
```

Re-assignment:

```
BASIC               Machine code
K=B                  LDA B
                     STA K
```

Adding constant:

```
BASIC               Machine code
B=B+17              1. LDA B
                       CLC
                       ADC # 17
                       STA B
                    2. LDA B
                       CLC
                       ADC # $11
                       STA B
```

Add and subtract:

```
BASIC               Machine code
B=B+K-127              LDA B
                       CLC
                       ADC K
                       SEC
                       SBC # $7F
                       STA B

B=B+K-S                LDA B
                       CLC
                       ADC K
                       SEC
                       SBC S
                       STA B
```

Multiplication:

```
BASIC               Machine code
N=N*2               1. LDA N
                       CLC
                       ADC N
                    2. ASL N
N=N*8                  ASL N
                       ASL N
                       ASL N

N=N*3                  LDA N
                       ASL A
                       CLC
                       ADC N
                       STA N
```

Incrementing:

```
BASIC                   Machine code
K=K+1                     INC K
K=K-1                     DEC K
```

Expressions

```
BASIC                   Machine code
K=8★(K+S)                 LDA K
                          CLC
                          ADC S
                          ASL A
                          ASL A
                          ASL A
                          STA K

S=B+(2★F+D)               LDA F
                          ASL A
                          CLC
                          ADC D
                          ADC B
                          STA S
```

Subroutine calls:

```
BASIC                   Machine code
GOSUB 2460                JSR SAVE
RETURN                    RTS
```

(Note that an arbitrary destination label can be used in assembly code but, in BASIC, we are restricted to a meaningless line number)

Simple loops:

Converting FOR/NEXT loops to machine code is straightforward except for the danger of being one out in the loop count. The following examples are merely to illustrate how the loop count can be set up. The actual process within the loop is left undefined but represented by "...".

```
BASIC                   Machine code
FOR N=1 TO 18             LDX   1
                          BACK ...
        ...               ...
        ...               ...
NEXT                      INX
                          CPX # $12
                          BNE BACK

FOR N=0 TO 90 STEP 3      LDX   0
                          BACK ...
        ...               ...
NEXT                      CLC
                          TXA
                          ADC # 3
                          TAX
                          CPX # 93
                          BNE BACK
```

(Note we have transferred X to A in order to add 3 each time round)

```
FOR N=S TO F-1            LDX S
        ...               BACK ...
        ...               ...
                          ...
NEXT                      INX
                          CPX F
                          BNE BACK
```

```
FOR N=5 TO 1 STEP -1      LDX # 5
        ...               BACK ...
        ...               ..
NEXT                      ...
                          DEX
                          BNE BACK
```

(Note that a decrementing loop is easier. It also saves a comparison instruction because a BNE test is sufficient.)

Conditional branching:

```
BASIC                   Machine code
IF K=0 THEN GOTO 500      LDA K
        ...               BEQ BLOGGS
        ...               ...
        ...               ...
NEXT                      BLOGGS ...
```

```
IF K<>0 THEN GOTO 500     LDA K
        ...               BNE BLOGGS
        ...               ...
        ...               ...
NEXT                      BLOGGS ...
```

```
IF K< 0 THEN GOTO 500     LDA K
        ...               BMI BLOGGS
        ...               ...
        ...               ...
500 ...                   BLOGGS ..
```

```
IF K> = 0 THEN GOTO 500   LDA K
        ...               BPL BLOGGS
        ...               ...
        ...               ...
500 ...                   BLOGGS ...
```

(Remember that Zero is recognised as a positive number)

```
IF N=0 THEN S=T           LDA N
                          BNE BLOGGS
                          LDA T
                          STA S
                          BLOGGS ...
```

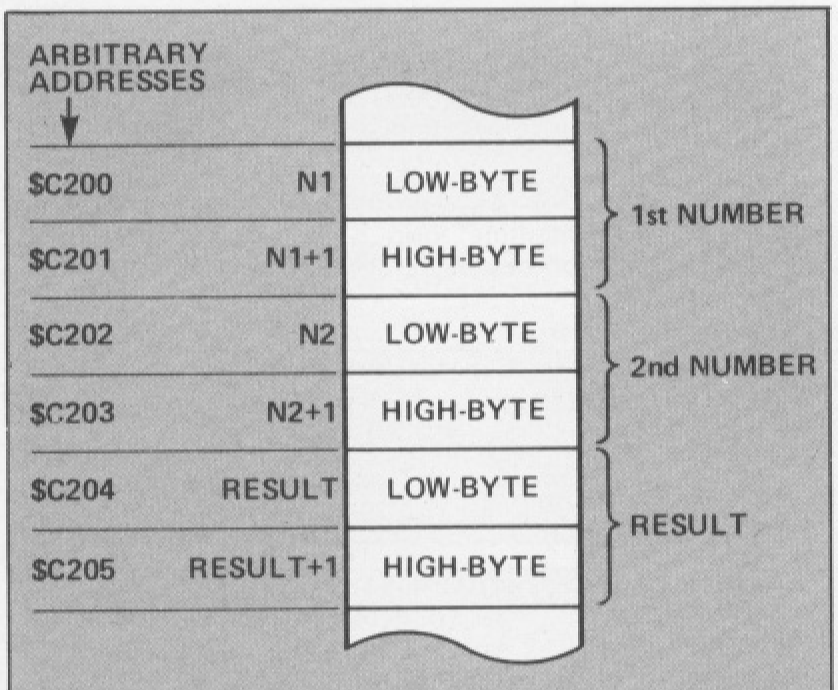
```
IF K=0 THEN T = -T        LDA K
                          BNE BLOGGS
                          LDA T
                          EOR # $FF
                          CLC
                          ADC # 1
                          STA T
                          BLOGGS ...

IF N=5 THEN GOTO 500      LDA N
        ...               CMP # 5
        ...               BEQ BLOGGS
500 ...                   ...
                          BLOGGS ...
```

```
IF N<>S THEN GOTO 500     LDA N
        ...               CMP S
        ...               BNE BLOGGS
        ...               ...
500 ...                   BLOGGS ...
```

Handling larger numbers

The previous examples have assumed numbers are within the capacity of a single byte. However, as we have mentioned in Part



Memory layout of double byte addition

Figure 7.1

5 of this series, we can handle larger numbers by using two or more bytes to handle each number. In machine code, this is awkward and long-winded rather than difficult. The following BASIC equivalents are using two bytes for a number. It is assumed that in memory, they occupy a pair of adjacent locations labelled, for example, NUMBER and NUMBER+1. The low-byte in NUMBER and the high-byte in NUMBER+1. We can, of course, use any other pair of symbolic addresses such as S and S+1, or K and K1.

Incrementing a double byte loop counter:

If a loop is to revolve more than 255 times, we must use a double byte loop counter. This entails knowing how to increment (or decrement) a double byte number. Study the following few lines, which should be visualised as the bottom of a processing loop. The top of the loop will be at the branch destination label 'BACK'.

```
INC  NUMBER
BNE  BACK
INC  NUMBER+1
BACK ...
```

Assume that NUMBER and NUMBER+1 both start at zero. While the count remains less than 255, only the low-byte is incremented because of the BNE branch to BACK. On the 256th revolution, NUMBER goes over the top to zero again. When this occurs, the BNE branch is not taken and the high-order byte NUMBER+1 is incremented. The inner loop revolves 256 times for each revolution of the outer loop. Since the outer loop can also revolve 256 times, the total number of revolutions possible by the combined loop is 256x256 = 65,536. However, if the loop count is to be some lesser value than this maxima, the locations NUMBER and NUMBER+1 must first be loaded with an offset value instead of zero.

Adding double byte numbers

```
BASIC                 Machine code
R = N1+N2               CLC
                        LDA N1
                        ADC N2
                        STA RESULT
                        LDA N1+1
                        ADC N2+1
                        STA RESULT+1
```

This should be studied with the aid of Figure 7.1 which shows how the memory locations should be visualised.
No provision is made for results which exceed the capacity of a two-byte number.

Subtracting double byte numbers

```
BASIC                 Machine code
R = N1-N2             SEC
                      LDA N1
                      SBC N2
                      STA RESULT
                      LDA N1+1
                      SBC N2+1
                      STA RESULT+1
```

Double byte conditional branching:

```
BASIC                 Machine code
IF N<>0 THEN GOTO 500 LDA N
        ...           BNE BLOGGS
        ...           LDA N+1
        ...           BNE BLOGGS
500     ...           ...
                      ...
                      BLOGGS ...
IF N = 0 THEN GOTO 500 LDA N
        ...           BNE T
        ...           LDA N+1
        ...           BEQ BLOGGS
500     ...           T...
                      ...
                      ...
                      BLOGGS
```

(Note that an intermediate branch to destination T is used for this test because, if the low-byte is non-zero, it is a waste of time testing the high-byte).

```
IF N1<>N2 THEN GOTO 500 LDA N1
        ...             CMP N2
        ...             BNE BLOGGS
        ...             LDA N1+1
        ...             CMP N2+1
500     ...             BNE BLOGGS
                        ...
                        BLOGGS ...
```

Printing characters

Printing characters on the screen is best carried out by making use of an operating system routine in the Kernal ROM called CHROUT, the details of which are as follows:

CHROUT
Function: send character to output channel.
Call address: $FFD2 (65490 decimal).
Parameter register: Accumulator must contain the character code before calling the routine.
Preparation routine: if the character is to be sent to any peripheral other than the screen, preparation is required by calls to CHKOUT and OPEN.

```
BASIC                 Machine code
PRINT "A"             LDA # 65
                      JSR $FFD2
```

Note: although, for simplicity, we have used an absolute address for calling the subroutine, we should point out that it is bad practice. The correct way, of course, is to assign it to a symbolic address at the top of the program.

NEWTRENDS TECHNOLOGY

HARDWARE

CBM 64 computer	196.00
SX 64 portable	850.00
VIC1541 disk drive	209.00
VIC1701 colour 14"	225.00
VIC1530 C2N cassette	39.95
MPS801 printer	210.00
MPS802 printer	325.00
DPS1101 daisy wheel	399.95
VIC1520 printer	95.95

SOFTWARE	*List Price*	*Our Price*
HOME & BUSINESS		
VIZAWRITE 64 (d)	79.95	69.95
VIZAWRITE 64 (c)	89.95	79.95
VIZASPELL 64 (d)	59.95	49.95
VIZAWRITE + VIZASPELL (d)	99.95	89.95
VIZASTAR 64 (d)	99.95	89.95
SUPERBASE 64	99.95	89.95

SAVE £10

(Applications software for Superbase 64)

STEPPING STONES

Accountants Time Recording	9.95
Solicitors Time Recording	9.95
Club Membership	9.95
Sales Daybook	9.95
Purchase Daybook	9.95
Cash Book	9.95
Estate Agents	9.95
Job Costing & Estimating	9.95
Stock Records	9.95
Villas & Apartments	9.95
Homebase 1	17.95
birthdays, freezer, sport results	
Homebase 2	17.95
bookcase, dirt, tv	
Homebase 3	17.95
hi-fi, holidays, sportstars	
Homebase 4	17.95
addresses, gardening, milkman	

PRACTICALC 64 (d or t)	44.50	41.50
PRACTIFILE 64 (d)	44.50	41.50
INVENTORY 64 (d)	29.95	26.95
PS programmable spreadsheet (d)		69.95

SAVE £3

COMMODORE

Easyscript	75.00	69.00
Easyspell	50.00	44.00
Easystock	75.00	70.00
Easyfile	50.00	45.00
Intro to Basic 1		14.95
Intro to Basic 2		14.95
Simons Basic (c)		50.00
Future Finance (d)		75.00
Petspeed Basic Compiler		50.00

ELECTRONIC AIDS (TEWKSBURY) LTD.
INTEGRATED ACCOUNTS 344.45
Very powerful customer supplier accounts (abe), printing, purchase & sales ledgers, day book, period VAT reports, full acct printout, own form invoice printing, full double entry nominal ledger to trial balance, P&L & balance sheet, single debit matching single credit and detailed audit trail printout to screen or hard copy for single transactions or all accounts. Write for brochure.

AUDIOGENIC

MICRO MAGPIE (d)	39.95
MICRO SWIFT (t)	19.95
MICRO SWIFT (d)	19.95
HOME OFFICE 64 (t)	14.95
MICRO WORDCRAFT (d)	24.95
MAGPIE	99.95
Forth	29.95
R NEST 12/6 (t)	6.95
Monitor	29.95

DATAVIEW WORDCRAFT

Wordcraft 64	49.95
PRECISION (d)	
Accountants Time Records	9.99
Cash Book	9.99
Club Membership	9.99
Estate Agency	9.99
Job Costing	9.99
Purchase Daybook	9.99
Sales Daybook	9.99
Solicitors Time Records	9.99
Travel Agency	9.99

HANDIC

Calc Result (easy)	49.95
Calc Result (advanced)	99.00
Diary 64	29.95
MON 64	39.95
Stat 64	29.95

ACCESSORIES (inc. p&p, VAT).
d = diskette c = cartridge t = tape
DISKETTES 5¼" (per 10)

	SS/SD	SS/DD	DS/DD
NEWTRENDS	13.50	16.00	18.50
CONTROL			
DATA	—	16.50	19.60
BASF	—	16.50	19.60
3M	—	16.60	20.00
DYSAN	—	25.00	35.00

CASSETTES — NEWTRENDS
— in pkgs of 10

C-10 £2.90
C-15 £3.00

NEW

REEL TO REEL
C-10 £4.50
C-15 £4.75

Diskette library case (holds 10 diskettes)	2.80
Lockable Case	
(40 diskettes)	18.50
(80 diskettes)	22.50

FAST LOADERS

RAM 1541 EXPRESS	49.95
ARROW C-64	26.00
plus C-64 (c)	44.00

PROGRAMMING AIDS

"INSIDE QUOTES"	1.00
Easy to use begininning programmers template w/instruction sheet.	
QUICK BROWN FOX	2.00
program template	
EASY SCRIPT	2.00
program template	
BASIC	2.00
teach yourself CBM Basic template	
PROGRAMMERS AID CARTRIDGE	2.00
template for the VIC—20	

INTERFACES

STACK RS-232	32.95
Centronics (t)	23.50
Centronics (d)	28.50
IEEE-488	59.00
VIZA CBM 64 to PET	33.95
Centronics cable	20.75

UTILITIES

THE QUILL (t)	14.95
THE QUILL (d)	19.95

EDUCATION

COMMODORE	English Language	9.95
	Biology	9.95
AUDIOGENIC	French (t,d)	12.95
	German (t,d)	12.95
	Italian (t,d)	12.95
	Spanish (t,d)	12.95

SUNSHINE

COMMODORE

For the Commodore 64:

Programming for Education on the Commodore 64 John Scriven and Patrick Hall	£5.95 net
Artificial Intelligence for the Commodore 64 Keith and Steven Brain	£6.95 net
Business Applications for the Commodore 64 James Hall	£5.95 net
Mathematics on the Commodore 64 Czes Kosniowski	£5.95 net
The Working Commodore 64 David Lawrence	£5.95 net
Commodore 64 Machine Code Master David Lawrence and Mark England	£6.95 net
Graphic Art for the Commodore 64 Boris Allan	£5.95 net
Commodore 64 Adventures Mike Grace	£5.95 net
Advanced Programming Techniques on the Commodore 64 David Lawrence	£5.95 net
DIY Robotics and Sensors for the Commodore 64 John Billingsley	£6.95 net
Machine Code Games Routines for the Commodore 64 Paul Roper	£6.95 net
Commodore 64 Disk Companion David Lawrence and Mark England	£7.95 net
Simulation Techniques on the Commodore 64 John Cochrane	£6.95 net
Machine Code Graphics and Sound for the Commodore 64 David Lawrence and Mark England	£6.95 net
Writing Strategy Games on your Commodore 64 John White	£6.95 net
Commodore 64 Music Ian Waugh	£6.95 net

For the Commodore C16 and Plus 4

The Working Commodore C16 David Lawrence	£6.95 net
The Commodore C16/Plus 4 Companion Brian Lloyd	£5.95 net

EXPORT ORDERS TAX FREE

DEALER ENQUIRIES WELCOME
write for details

Please add 50p for orders under £20.00 (Europe add £1). Make cheques and P.Os to: **NEWTRENDS TECHNOLOGY.** Please allow two weeks for delivery

35 North Audley Street, London W1Y 1WG. Telex: 24828 PROMOV

NEWTRENDS TECHNOLOGY

I enclose cheque/P.O. for £ ....

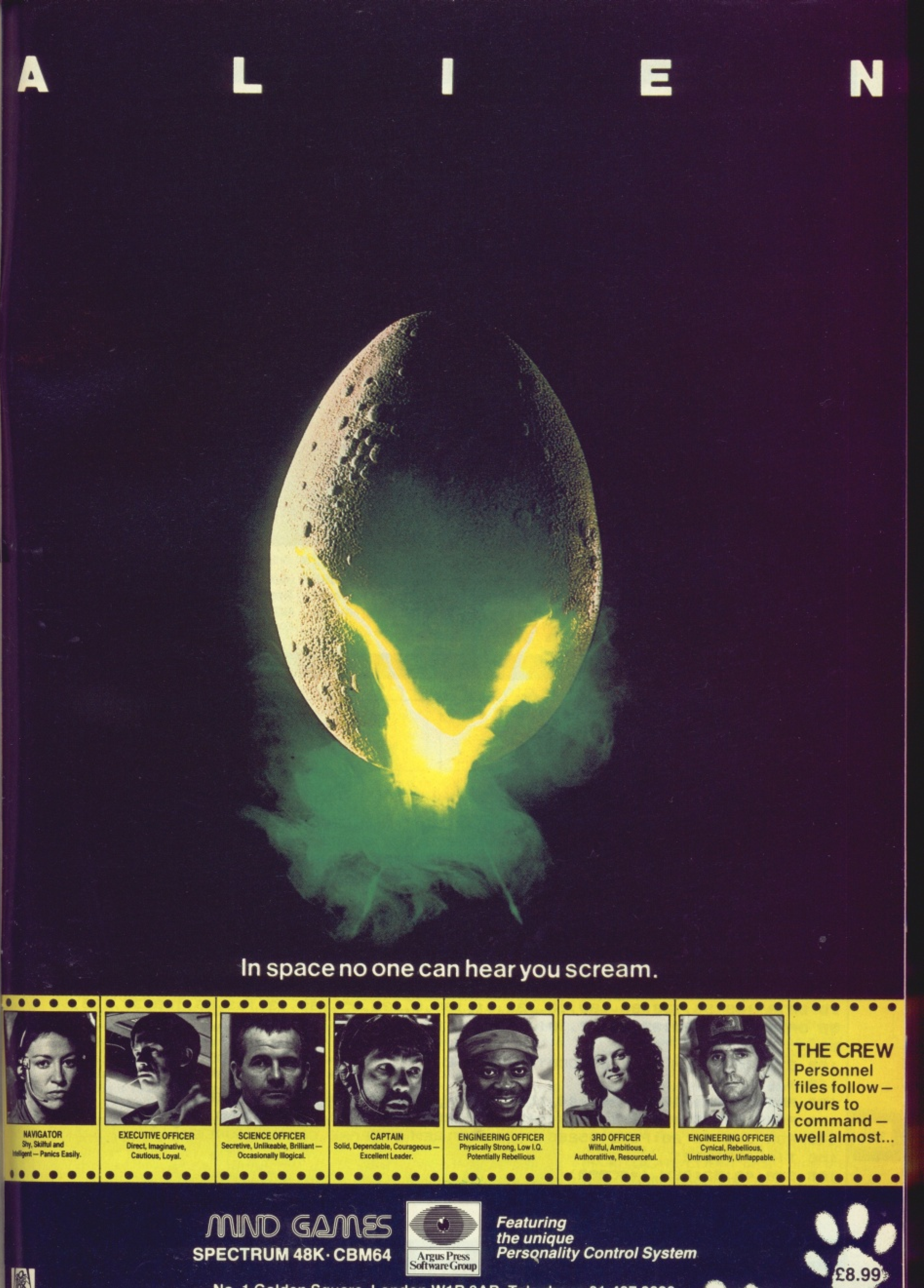
Please send me .... copies of "INSIDE QUOTES"

NAME ....

ADDRESS ....

....

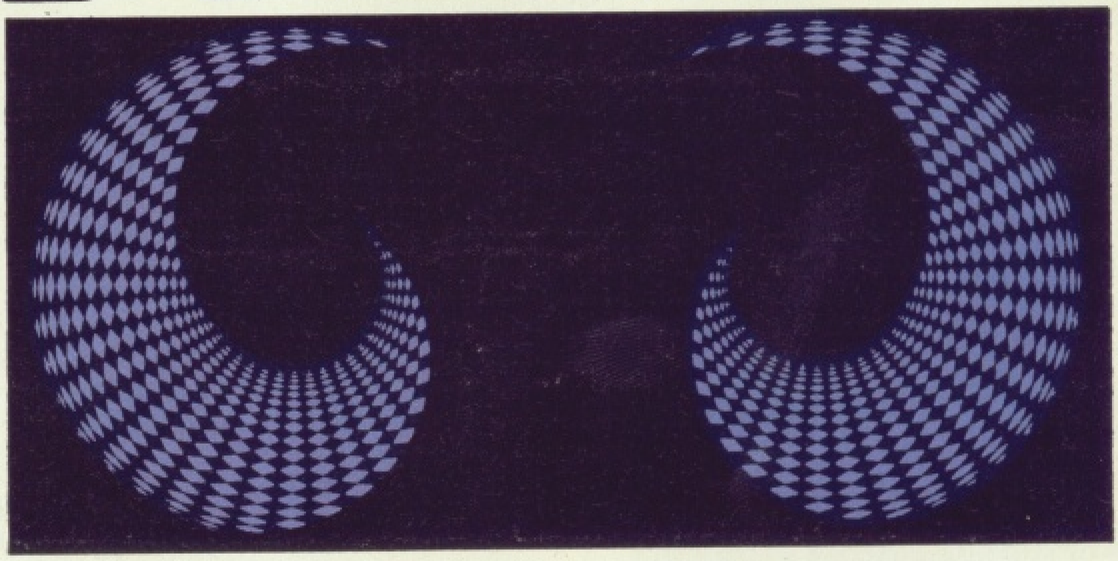
.... POST CODE ....



ALIEN
In space no one can hear you scream.
NAVIGATOR
Shy, Skilful and Inteligent — Panics Easily.
EXECUTIVE OFFICER
Direct, Imaginative, Cautious, Loyal.
SCIENCE OFFICER
Secretive, Unlikeable, Brilliant — Occasionally Illogical.
CAPTAIN
Solid, Dependable, Courageous — Excellent Leader.
ENGINEERING OFFICER
Physically Strong, Low I.Q. Potentially Rebellious
3RD OFFICER
Wilful, Ambitious, Authoratitive, Resourceful.
ENGINEERING OFFICER
Cynical, Rebellious, Untrustworthy, Unflappable.
THE CREW
Personnel files follow — yours to command — well almost...
MIND GAMES
SPECTRUM 48K · CBM64
Argus Press Software Group
Featuring the unique Personality Control System
£8.99
No. 1 Golden Square, London W1R 3AB. Telephone

RELO

M.C. Hart shows you how to save machine code at the end of a BASIC program and then use an internal ROM routine to locate it.



ONCE A MACHINE CODE ROUTINE HAS been written it may be loaded back into its original position (absolute address) with the command LOAD "PROGRAM", devie number, 1. To activate the routine one must be aware of the entry point of the machine code and access it with the relevant SYS call. Even experienced users of machine code can still forget the entry points to their routines and the problem is confounded if a routine has been written for use by an inexperienced user.

To overcome these problems, machine code routines could be written so that they 'sat' at the end of a normal BASIC program. This is quite easy to achieve by altering the pointers to the end-of-program (=start-of-variables) so that they include the machine code as well as the BASIC program. For the C-64 (as well as the VIC) these pointers are at \$2D-\$2E (45-46 decimal). The BASIC program could then point to another piece of machine code which could relocate the original piece of code and run it as well if necessary. The user would simply need to LOAD the program and then RUN it.

The CBM interpreter actually has its own 'memory move' routine built into it. The interpreter calls this into play automatically once you insert a piece of code into an existing BASIC program but

Program Listing 1

```
1 REM *** RELOCATE/COUNT DEMO ***
2 REM ***        FOR C-64         ***
3 :
4 REM ***     BY M. C. HART       ***
5 :
6 :
10 FOR J=0 TO 35:READ X:POKE 3328+J,X:NEXT:REM COUNT DEMO (3328 = $0D00)
20 DATA 162,5,169,48,157,15,4,202,16,250,162,5,189,15,4,24,105,1,201,58,240,5
30 DATA 157,15,4,208,239,169,48,157,15,4,202,16,233,96
40 :
50 :
60 FOR J=0 TO 27:READ X:POKE 3072+J,X:NEXT:REM RELOCATOR DATA (3072 = $0C00)
61 REM RELOCATES FROM $0D00-$0D23
62 REM (=3328-3363) TO NEW LOCATION I.E.
63 REM $C000 (=49152) - SEE LINE 160
64 :
70 DATA 169,36,133,90,169,13,133,91,169,0,133,95,169,13,133,96,169,36,133,88
80 DATA 169,192,133,89,32,191,163,96
90 :
100 SYS 3072:REM CALL RELOCATE ROUTINE
110 :
120 PRINTCHR$(147);CHR$(144):REM CLEAR SCREEN;BLACK CHARACTERS
130 POKE 53280,15:POKE 53281,15:REM GREY 3 BORDER/BACKGROUND
140 FOR J=0 TO 20:POKE 55296+J,0:NEXT:REM PREPARE COLOUR SCREEN
150 TI$="000000"
160 SYS 49152:REM (=$C000)
170 T=TI/60:PRINT:PRINT"TIME="T"SECS"
```

CATOR

Program Listing 2

```
B*        RELOCATOR ROUTINE
.;
.
0C00 A9 24         LDA #$24
0C02 85 5A         STA $5A
0C04 A9 0D         LDA #$0D
0C06 85 5B         STA $5B
0C08 A9 00         LDA #$00
0C0A 85 5F         STA $5F
0C0C A9 0D         LDA #$0D
0C0E 85 60         STA $60
0C10 A9 24         LDA #$24
0C12 85 58         STA $58
0C14 A9 C0         LDA #$C0
0C16 85 59         STA $59
0C18 20 BF A3      JSR $A3BF
0C1B 60            RTS
```

Program Listing 3

```
B*            COUNT ROUTINE
.;
.
0D00 A2 05            LDX #$05
0D02 A9 30            LDA #$30
0D04 9D 0F 04         STA $040F,X
0D07 CA               DEX
0D08 10 FA            BPL $0D04
0D0A A2 05            LDX #$05
0D0C BD 0F 04         LDA $040F,X
0D0F 18               CLC
0D10 69 01            ADC #$01
0D12 C9 3A            CMP #$3A
0D14 F0 05            BEQ $0D1B
0D16 9D 0F 04         STA $040F,X
0D19 D0 EF            BNE $0D0A
0D1B A9 30            LDA #$30
0D1D 9D 0F 04         STA $040F,X
0D20 CA               DEX
0D21 10 E9            BPL $0D0C
0D23 60               RTS
```

it is useful to exploit it for our own purposes. It will handle non-overlapping moves both forwards and backwards with no problems and will also move code down (i.e. later) in memory even with overlap. It fails if the new code location overlaps with the existing code location and is *before* it in memory, so it is best avoided in these circumstances.

In order to illustrate the principles involved, the RELOCATE/COUNT program reads in the machine code for a program called COUNT into space after the BASIC program itself. The BASIC program starts at $0800 and ends at about $0B77 (depending on how many spaces are put in the program) so the code read in by lines 10-30 (into $0D00 and following) is well clear. Now a second piece of code is read in (lines 60-80) and this is the 'RELOCATOR' code itself — it is read into $0C00. There is nothing sacred about this particular location, however, and it could equally well have gone into the cassette buffer ($033C= 828 decimal). Line 100 calls the RELOCATOR routine which copies the first piece of machine code from $0D00-$0D23 into $C000-$C023. Lines 120-150 prepare the screen for what is to follow and the relocated code is called in Line 160.

I have been deliberately vague about what COUNT actually does as it is a bit of a surprise and is well worth watching if only to illustrate the speed of machine code. It is not original but is adopted from work previously published by Mike Gross-Niklaus. The real problem is how to relocate any piece of machine code rather than this particular demo.

LISTING 2 is the machine code disassembly for the RELOCATE routine.

End of present routine	(+1) into $5a-$5B	(ditto for VIC,$57-$58 in PETS
Start of present routine	into $5F-$60	(ditto for VIC, $5C-$5D in PETS)
End of relocated routine(+1)	into $58-$59	(ditto for VIC, $55-$56 in PETS)
Relocate routine	($A3BF (C-64); $C3BF (VIC);	$C2DF (BASIC 2); $B357 (BASIC 4)

The parameters that require passing are as follows, in each case low byte followed by high byte in conventional 6502/6510 fashion:

After the RELOCATE call you may insert your own call to 'turn on' your own routine if required. You could also NEW your driver program and return to BASIC is necessary although this is probably easier from the BASIC driver. Your BASIC driver can be as primitive as a single SYS call to the RELOCATOR routine although, of course, you would have to ensure that the end-of program pointers were adjusted as previously described. I use the TRANSFER command in SUPERMON to put the code from where it normally resides to the end of a BASIC driver.

LISTING 3 is the machine code disassembly for the COUNT demo itself. Machine code fanatics can puzzle out for themselves how it works.

Even if machine code is a complete mystery to you, I am sure that you will find the RELOCATE/COUNT BASIC program well worth running just for its demonstration effect alone.

YOUR COMMODORE

Lineage: 40p per word.
Semi display: £9.00 per single column centimetre
Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01-437 0699

Send your requirements to:
BECKI WILSON
ASP LTD, 1 GOLDEN SQUARE,
LONDON W1

SOFTWARE

COMMODORE 64 SOFTWARE
GAMES AND BUSINESS USE

Also new releases list every 2 weeks. Send your name and address to go onto our mailing list to:

M. J. Seaward (Dept Y.C.O.)
St Olaf's Road, Hill Head,
Stratton, Nr. Bude, Cornwall.
Tel: Bude 4179 up to 10 pm.

ELECTRONIC AIDS SOFTWARE
Payroll and Accounting software for COMMODORE 64

0386 49339

SOFTWARE APPLICATIONS

HOME ACCOUNTS. Put your house in order! Probably home computing's best use! Comprehensive coverage of bank accounts. Credit cards, HP inbuilt accuracy check. Records all transactions. Projects cashflow for any period ahead. Available for C.16 CBM64 or VIC-20. £8.45 or *free* details from Discus Software, Freepost, Windmill Hill, Brixham TQ5 9BR. Telephone 08045 55532.

FOR HIRE

VIC-20 / CBM-64 SOFTWARE HIRE
NO MEMBERSHIP FEE

Send 2 x 17p stamps for your hire kit to:
VSH (YM), 242 RANSOM ROAD, MAPPERLEY, NOTTINGHAM.
Please state which machine.

LIBRARIES

COMMODORE 64
MIDLAND COMPUTER LIBRARY

The first and the largest Commodore 64 library in the world. Often copied but never equalled.

1) All the latest titles on cassette, disc and cartridge.
2) Hire your first two games absolutely free.
3) Life membership £6.
4) Hire charge from only £1 inc p&p for full 7 days.
5) No limit to the amount of games you can hire.
6) All games originals with full documentation.
7) Games, Business and Educational software.
8) Return of post service
9) Lowest new hard and software prices in the UK.
10) Join now on 14 day money back guarantee.

Over 950 different titles in stock. Send cheque or postal order for £6 with your two selections or large SAE for further details to: **Midland Computer Library, Dept Y.C., 172 Studley Road, Redditch, Worcs.**
TELEPHONE: 0527 23584

ADVERTISE YOUR COMMODORE PRODUCTS HERE AND REACH 58,000 COMMODORE USERS.

COMMODORE 64 SOFTWARE LIBRARY
Over 200 titles. 2 weeks hire £1.00 Membership £5.00. Stamp for list.
Les Wilson (CY), 100 Blenheim Walk, Corby, Northants.

YOUR COMMODORE — CLASSIFIED ADVERTISEMENT — ORDER FORM

Advertise nationally in these columns for only 40p per word (min charge 15 words). Simply print your message in the coupon and send with your cheque or postal order made payable to **Argus Specialist Publications Ltd** to:

CLASSIFIED DEPARTMENT
YOUR COMMODORE
No 1 Golden Square, London W1
Tel: 01-437 0699

Name ....

Address ....

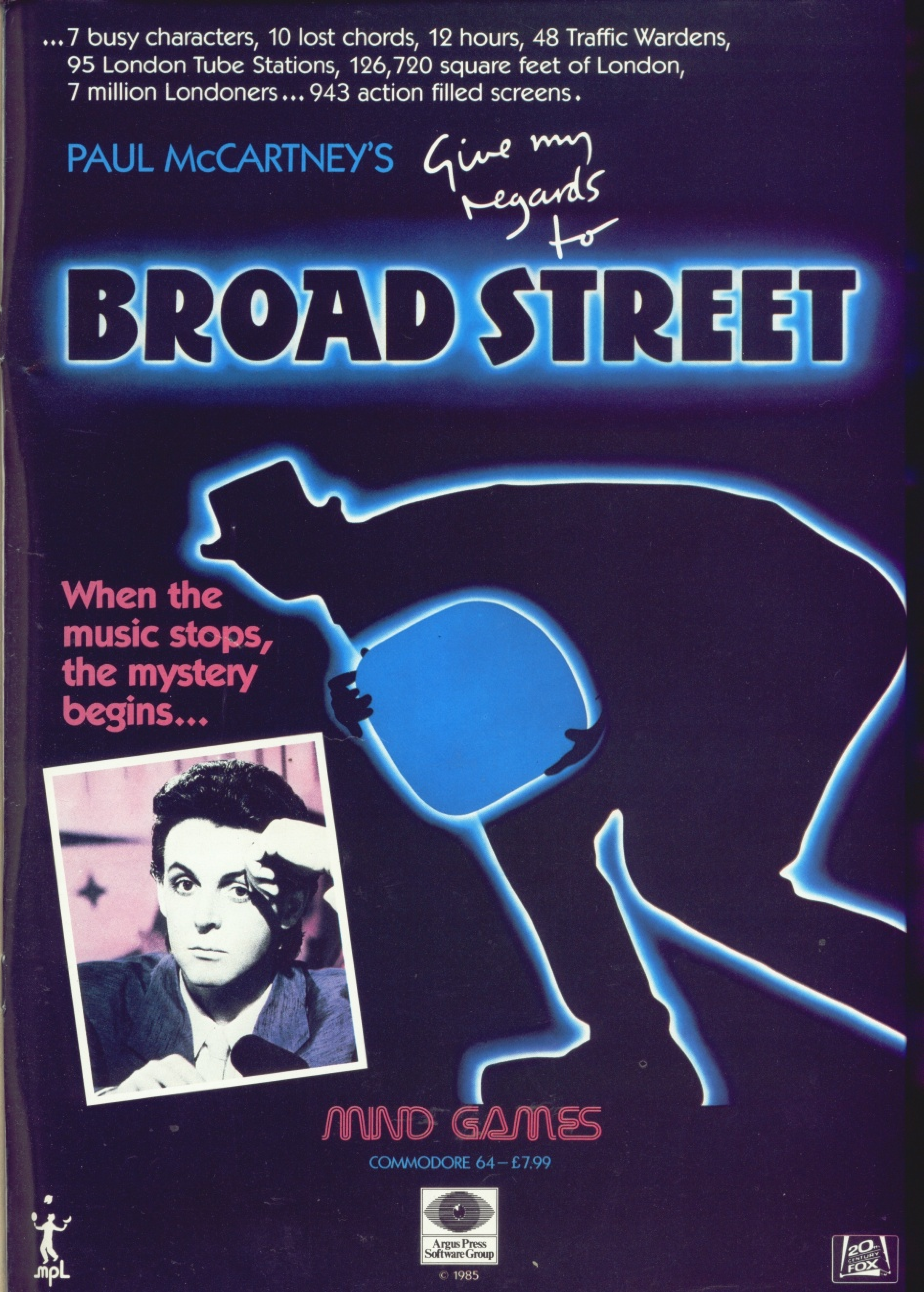
....

....

Tel no (day) ....

Please place my advert in **YOUR COMMODORE** for .... issues.

Please indicate no. of insertions required.


...7 busy characters, 10 lost chords, 12 hours, 48 Traffic Wardens, 95 London Tube Stations, 126,720 square feet of London, 7 million Londoners... 943 action filled screens.
PAUL McCARTNEY'S Give my regards to
BROAD STREET
When the music stops, the mystery begins...
MIND GAMES
COMMODORE 64 – £7.99
Argus Press Software Group
© 1985
mpL
20th CENTURY FOX

Choosing the right computer is a good start — but can you find the right software?



At SUPERSOFT we're very conscious of the fact that people who spend several hundred pounds on computer equipment are looking to do rather more than play Space Invaders.

Financial planning is a rather grand name for something you've been doing all your life – making ends meet! Perhaps if Mr Micawber had used **BUSICALC** he would have been able to balance the books a little better.

For home, club or small business use BUSICALC 1 should pay for itself in no time at all; for larger companies we recommend BUSICALC 3, one of the few really valuable programs that you can learn to use in a day.

Although your Commodore 64 is a powerful musical instrument you need to be a pretty good programmer to understand how it all works. Unless, of course, you buy **MUSIC MASTER!**

To use MUSIC MASTER requires no prior musical knowledge, though in the hands of an experienced musician it will prove an invaluable tool. You don't need to know the first thing about programming either! MUSIC MASTER is the musical equivalent of a word processor, remembering the notes you play and allowing you to replay and edit them as you wish.

INTERDICTOR PILOT is a space flight simulator. Nowadays simulators are widely used to train pilots and astronauts because – to be frank – it's a lot cheaper (and safer) than the real thing!

Imagine, if you will, life in the 22nd century: space travel is commonplace, and on the outskirts of the galaxy the first war between civilizations is being fought. A shortage of trained pilots has prompted the Federation to develop a computer simulation that allows raw recruits to gain experience without paying for their mistakes with their lives. With the aid of your Commodore 64 you too can learn to pilot the Interdictor Mk 3 craft. But be warned – this is no game!

SUPERSOFT

SUPERSOFT, Winchester House, Canning Road, Wealdstone, Harrow, Middlesex HA3 7SJ
Telephone: 01-861 1166

Other SUPERSOFT products include the **MIKRO ASSEMBLER** cartridge, the only assembler that's ideal for beginners yet powerful enough for the professional (most of our competitors use it!). The **VICTREE** cartridge adds dozens of commands to Basic including toolkit aids and disk commands; or on disk there's **MASTER 64,** a really comprehensive package for the keen programmer.

Of course, we do also publish games programs, and with classics like **STIX, QUINX** and **KAMI-KAZE** in our range we are one of the market leaders. But we most enjoy coming up with the sort of programs that are going to be in use for months and years, not hours and days – the sort of programs that make you glad that you bought a computer – and glad that you bought SUPERSOFT!

You won't find SUPERSOFT products on the shelves of your local supermarket. But most specialist shops stock titles from our extensive range (and are prepared to obtain other programs to order). However you can also buy direct by sending a cheque (pre-paid orders are post free!), by calling at our offices, or over the telephone using your ACCESS card.